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

Bakalářská práce

Zařízení pro měření a sběr dat diabetického pacienta

Plzeň 2021 Radek Miklánek

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Radek MIKLÁNEK**
Osobní číslo:	**A17B0292P**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Informatika**
Téma práce:	**Zařízení pro měření a sběr dat diabetického pacienta**
Zadávající katedra:	**Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se s nemocí diabetes mellitus a s druhy biomedicínských veličin relevantních k dynamice glykémie. Dále se seznamte s programátorským rozhraním systému SmartCGMS (https://diabetes.zcu.cz/smartcgms).
2. Analyzujte dostupné senzory těchto veličin, možnosti jejich integrace do jednoho zařízení a možnosti přenosu měřených hodnot na sběrné zařízení (např. mobilní telefon).
3. Navrhněte zařízení složené z již hotových modulů, které vybrané senzory integruje a měřená data přenáší navrženým síťovým protokolem do sběrného zařízení. Při návrhu zohledněte energetickou náročnost.
4. Sestrojte prototyp tohoto zařízení, implementujte pro něj příslušný software a vytvořte knihovnu pro systém SmartCGMS pro sběrné zařízení, díky které je možné tato data číst a ukládat.
5. Otestujte celé řešení z hlediska funkčnosti a zhodnoťte dosažené výsledky.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Ing. Martin Úbl**
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **5. října 2020**
Termín odevzdání bakalářské práce: **6. května 2021**

L.S.

Doc. Dr. Ing. Vlasta Radová
děkanka

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 26. října 2020

Prohlášení

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

V Plzni dne 5. května 2021

Radek Miklánek

Abstract

The objective of this bachelor thesis is to analyse measurands which affects diabetic patients. On the basis of this analysis suitable sensors capable of obtaining the variables of these measurands were selected. All sensors are controlled by a device that is obtaining data from the sensors. The obtained data are sent by wireless technologies. One of the objectives in the device design and construction was to achieve the lowest possible electric power consumption.

Another objective of this bachelor thesis is to design and implement a protocol for transmission of measurands to the mobile device. This transmission is provided by wireless technologies. When implementing the protocol, the emphasis is primary on modularity, so that the protocol can still be used when a sensor is added or removed.

The last objective of this thesis is to implement a library for the Smart-CGMS system. The library is used by a collection device that is able to read and store the measured data.

Abstrakt

Cílem této práce je analyzovat měřitelné veličiny ovlivňující diabetického pacienta a na základě této analýzy vybrat vhodné senzory schopné získat hodnoty těchto veličin. Senzory budou řízeny zařízením, které bude ze senzorů získávat data. Získaná data se budou odesílat pomocí některé z dostupných bezdrátových technologií. U zařízení je vyvinuta snaha o co nejmenší spotřebu el. energie.

Další důležitou součástí této práce bude návrh a implementace protokolu pro přenos naměřených hodnot do mobilního zařízení pomocí přenosových bezdrátových technologií. U implementace protokolu je kladen důraz především na modularitu, aby při případném přidání nebo odebrání senzoru stále bylo možné protokol použít.

Poslední částí práce je implementace knihovny pro systém SmartCGMS pro sběrné zařízení, které bude schopné číst a ukládat data.

Poděkování

Rád bych poděkoval Ing. Martinu Úblovi za odborné vedení a cenné rady v průběhu této práce.

Obsah

1 Úvod **10**
1.1 Diabetes mellitus 11
1.2 SmartCGMS 12

2 Analýza **15**
2.1 Měřitelné veličiny 15
2.1.1 Pohyb 15
2.1.2 Teplota těla 16
2.1.3 Okolní teplota 16
2.1.4 Krevní tlak 17
2.1.5 Stres 17
2.2 Přenos dat 18
2.2.1 IEEE 802.11 18
2.2.2 IEEE 802.15.4 19
2.2.3 Bluetooth 20
2.2.4 Mobilní síť 22
2.2.5 Shrnutí 24
2.3 Platforma měřicího zařízení 27
2.3.1 Arduino 27
2.3.2 Espressif ESP 28
2.3.3 Texas Instruments 28
2.3.4 STM32 29
2.3.5 Shrnutí 29
2.4 Senzory 32
2.4.1 Akcelerometr s gyroskopem 32
2.4.2 GPS modul 33
2.4.3 Měření teploty 34
2.4.4 Měření srdečního tepu 35

3 Návrh **37**
3.1 Měřicí zařízení . . . 37
3.1.1 Firmware . . . 37
3.1.2 Detektor pohybu . . . 38
3.1.3 Měření teploty . . . 40
3.1.4 Lokalizace . . . 40
3.1.5 Přenos dat . . . 40
3.1.6 Schéma zapojení . . . 41
3.2 Přenos dat do sběrného zařízení . . . 43
3.2.1 Hlavička protokolu . . . 44
3.2.2 Zahajovací zpráva . . . 44
3.2.3 Zpráva s požadavkem . . . 45
3.2.4 Informativní zpráva . . . 45
3.2.5 Zpráva nastavení . . . 46
3.2.6 Datová zpráva . . . 48
3.2.7 Datová zpráva se všemi daty z bufferu . . . 48
3.2.8 Ukončovací zpráva . . . 49
3.2.9 Zpráva kladného potvrzení . . . 49
3.2.10 Zpráva záporného potvrzení . . . 49
3.2.11 Stavový diagram připojení . . . 49
3.3 Aplikace sběrného zařízení . . . 51
3.3.1 Připojení k měřicímu zařízení . . . 51
3.3.2 Modul pro SmartCGMS . . . 52
3.3.3 Ukládání přijatých dat . . . 52
3.4 Filtr pro ukládání dat do SmartCGMS . . . 52
3.5 Protokol pro přenos dat do SmartCGMS . . . 52
3.5.1 Hlavička protokolu . . . 53
3.5.2 Datová zpráva . . . 53

4 Implementace **55**
4.1 Firmware pro měřicí zařízení . . . 55
4.1.1 Jádro . . . 57
4.1.2 Protokol . . . 57
4.1.3 Moduly . . . 60
4.2 Aplikace sběrného zařízení . . . 66
4.3 Integrace se SmartCGMS . . . 67
4.3.1 Protokol . . . 67

5 Testování **69**
5.1 Celková funkcionalita 69
5.2 Test spotřeby energie 71
5.3 Přesnost měření teploty prostředí 72
5.4 Přesnost měření tělesné teploty 75
5.5 Ověření přesnoti pohybového modulu 77
5.6 Ověření přesnosti GPS modulu 80

6 Závěr **82**

Literatura **84**

Seznam zkratek **88**

Seznam obrázků **91**

Seznam tabulek **92**

Obsah přiloženého CD **92**

Příloha A Instalace software **94**

Příloha B Přenos dat do sběrného zařízení **96**

Příloha C Protokol pro přenos dat do SmartCGMS **102**

1 Úvod

V posledních letech narůstá celosvětově počet pacientů s nemocí diabetes mellitus. Jen v České republice bylo k roku 2015 evidováno 858 010 pacientů [25] a mimo to jsou tu i lidé, kteří o této nemoci ani netuší, že jí vůbec mají.

Nemoc diabetes mellitus se dělí na hlavní dva typy [17]. Oba typy pacientů musí přizpůsobit svůj životní styl a podrobovat se pravidelným lékařským kontrolám. Průběh diabetu je ovlivněn širokou škálou aspektů, z nichž některé lze pozorovat měřením přidružených veličin. Tato práce se dále zabývá návrhem zařízení, které umožňuje tyto veličiny měřit a odesílat naměřené hodnoty do sběrného zařízení.

Takové zařízení by pak mohlo z dlouhodobého hlediska usnadnit práci lékařům nebo rovnou samotným pacientům. Lékařům by pomohlo například to, že by měli možnost vidět různé veličiny v průběhu času od poslední kontroly. Pacientům by naopak mohlo pomoci s jejich každodenním životem. Pomoc by byla zprostředkována skrze získaná data z měřených veličin. Na základě těchto dat, by si pak mohl pacient například plánovat svoje aktivity, nebo by byl včas upozorněn, že se může nacházet ve stavu, který pro něj může být nebezpečný. Posledním aspektem je, že shromážděná data ze zařízení mohou pomoci při výzkumu léčby diabetu.

Cílem této práce je vytvořit takové zařízení, které bude dostatečně modulární a bude umět komunikovat se senzory k němu připojenými. Data se budou předávat pomocí některé bezdrátové technologie do sběrného zařízení, kterým může být například mobilní telefon.

Pro přenos dat bude navržen a implementován síťový protokol. Bude v něm kladen důraz na možnou variabilitu přenášených údajů ze senzorů. U zařízení se předpokládá, že senzory bude možné přidat a odebrat.

Jako protikus k měřicímu zařízení bude vytvořena jednoduchá testovací apli-

kace pro mobilní zařízení. Cílem aplikace bude otestovat funkčnost navrženého měřicího zařízení. Aplikace bude rovněž předávat data do systému SmartCGMS.

1.1 Diabetes mellitus

Diabetes mellitus je skupina metabolických onemocněních projevujících se zvýšenou hladinou glokózy v krvy (*hyperglykémii*). Vysoká hladina glukózy v krvi je nejčastěji zapříčiněna nedostatečnou produkcí inzulinu nebo nízkou citlivostí buněk. Nemoc se dělí na několik typů:

- **I.typ** - V tomto případě dochází u pacientů k nedostatečné produkci inzulinu, kvůli poškození tzv. β-buněk [14]. Pacienti s tímto typem musí celoživotně dodávat tělu inzulin. Zároveň je třeba monitorovat hladinu glukózy v krvi a tu správně regulovat.
- **II. typ** - Oproti prvnímu typu mají pacienti s tímto typem diabetu inzulinu dostatek, ale jejich tělo ho nedokáže správně zpracovat, to vede k nadbytku inzulinu [14]. Tělo pacientů s tímto typem diabetu může inzulin stále produkovat. Léčba inzulinem v tomto případě nemusí být na místě. Místo toho se provádí léčba dietami a pacientům jsou předepisovány léky zvané *antidiabetika*.
- **Ostatní typy** - Ostatní typy diabetu se souhrně označují *sekundární diabetes*. Tento typ diabetu vzniká jako sekundární efekt jiných nemocí nebo léků, které jsou při nich podávány. Rovněž se může diabetes vyvinout jako průvodce těhotenství (gestační diabetes mellitus).

Nemoc diabetes přináší dva akutní rizikové zdravotní stavy. Jeden z nich nástává při nízké hladině glukózy v krvi a nazývá se *hypoglykémie*. Druhý nastává naopak, když je příliš vysoká hladina glukózy v krvi a nazývá se *hyperglykémie*. V obou případech je potřeba tento stav řešit okamžitou lékařskou pomocí. Akutní rizikové stavy vedou ve stavy chronické např. diabetická noha nebo neuropatie. Při vzniku onemocnění diabetické nohy dochází k infekci, vředům nebo odumírání tkání na noze. Neuropatie je onemocnění při kterém dochází k poškození nervů a může vést až ke špatné hybnosti končetin.

Na průběh glykemie může mít vliv řada aspektů a to strava, fyzická aktivita, rozdíl teplot, tlak , tep, stres, a další [23] [27] [1] [17].

1.2 SmartCGMS

Architektura SmartCGMS se sestává z několika filtrů propojených za sebou [29]. Všechny filtry mají stejné rozhraní. Rozhraní umožňuje komunikaci filtrů, kdy každý má jiný účel. Filtr může dělat například jednu z následujících činností:

- Čtení dat z databáze
- Čtení dat ze senzoru
- Výpočet dávkování inzulinu
- Výpočet podpůrných veličin
- Vykreslování grafů

Několik sestavených filtrů za sebou nedělá nic jiného, než že na vstupu přijímá událost, tu zpracuje a posílá jí dalšímu filtru. Událost se pak zpracovává postupně všemi filtry tak, jak jdou za sebou. Události může generovat i samotný filtr. Nemusí posílat jen ty, které mu přijdou na vstup. Zmiňovaná událost vyobrazená ve výpisu 1.1 je datová struktura. Ve struktuře je nejdůležitější následující výčet informací [19]:

- **Kód události** - Jednoznačný identifikátor události. Některé kódy slouží pro přenos informací a jiné pro ovládání filtru.
- **ID zařízení** - Jednoznačný identifikátor zařízení, které vyrobilo událost.
- **ID signálu** - Identifikátor typu veličiny. Například predikovaná hodnota glukózy v krvi, spočítaná hodnota glukózy v krvi, vypočítáná dávka inzulinu a další.
- **Čas zařízení** - Časová značka představující reálný čas.
- **Logický čas** - Časová značka pro určení pořadí, v jakém události vznikly.

- **Hodnota dat** - Může obsahovat různé typy hodnot podle *kódu události*.

Výpis kódu 1.1: Struktura události [19]

```
struct TDevice_Event {
   NDevice_Event_Code event_code;
   GUID device_id;
   GUID signal_id;
   double device_time;
   int64_t logical_time;
   uint64_t segment_id;

   union {
      double level;
      IModel_Parameter_Vector* parameters;
      double[] wstr_container* information;
   };
};
```

Síla architektury sestavající se z filtrů je velmi jednoduchá změna funkčnosti celku bez nutnosti upravovat zdrojový kód. Filtry se dají kdykoliv přeuspořádat a tím se upraví funkce celku. Například bude vytvořena řada filtrů, která bude umět na základě získáné hodnoty glukózy z krve dávkovat inzulin pomocí inzulinové pumpy. Budeme-li chtít přidat ukládání dat do databáze, přidáme na vhodné místo v řadě filtrů filtr, který toto zprostředkuje.

Filtr je entita implementující rozhraní `scgms::IFilter`. Toto rozhraní má dvě hlavní metody. První metodou je metoda `Do_Configure`, která je volána jednou při inicializaci řetězu filtrů. Díky této metodě je možné předat sadu parametrů, kterým je běh filtru přizpůsoben požadavkům. Druhou metodou je metoda `Do_Execute`, která je volána vnějším kódem vždy, když přijde nová událost od předchozího filtru v řadě.

Mezi ostatní entity systému patří například model, aproximátor, signál, solver nebo metrika.

Model je entita, která obsahuje logiku matematického modelu, například metabolismu člověka nebo modelu léčby. Aproximátor je entita, díky které lze aproximovat diskrétní hodnoty spojitou funkcí, a tím umožnit např. fun-

gování vybraným matematickým modelům, které vyžadují spojitý signál. Signál je entita, která slouží jako přepravka pro hodnoty signálu. Solver je algoritmus pro hledání parametrů matematického modelu. Metrika je entita, která zaštiťuje výpočet metriky referenčního signálu vůči vypočtenému. Tyto entity nejsou pro tuto práci relevantní a tak se jimi práce dále nezabývá.

Každá entita má přidružený tzv. *deskriptor*. To je struktura, která obsahuje informace o entitě, jako je například její ID, název, konfigurační parametry a jiné.

Všechny entity jsou implementovány v rámci sdílených knihoven, které exportují funkce pro jejich vytvoření a získání jejich deskriptorů. Pro filtry jsou to funkce `do_create_filter` a `do_get_filter_descriptors`.

Software SmartCGMS je možné zprovoznit i na nízkopříkonových zařízeních [19]. Je to dáno tím, že architektura má nízké nároky na výpočetní výkon, paměť a jiné systémové prostředky. Software je implementován pomocí jazyka C++ [19].

2 Analýza

V následující kapitole bude provedena analýza měřitelných veličin, přenosových bezdrátových technologií, platforem pro měřicí zařízení a senzorů.

2.1 Měřitelné veličiny

V této sekci budou probrány veličiny mající nepřímý vliv na diabetes. Mezi přímé veličiny patří například hladina glukózy v krvi. Tato veličina se běžně měří invazivní metodou. To znamená, že je potřeba zavést senzor do lidského těla a kvůli tomu by byly kladeny vyšší nároky (certifikace, regulace,...) na vytvářené měřicí zařízení, aby ho bylo možné využívat.

Pro veličiny z této analýzy budou vybrány vhodné senzory pro jejich měření.

2.1.1 Pohyb

Pohybová aktivita by se měla týkat nejen osob s diabetem, ale i ostatních, kdy je doporučováno mít pohyb alespoň pár hodin týdně. U diabetiků s typem II je to však ještě potřebnější. Diabetes často bývá doprovázen obezitou, poruchami metabolismu, zvýšenou pravděpodobností srdeční příhody, či cévní mozkovou příhodou. [31].

Pro tyto důvody je pohybová aktivita diabetikům doporučována, protože lze předejít výše zmíněným zdravotním problémům [31]. Kromě toho se při pohybu zvyšuje citlivost inzulinových receptorů a to vede ke snížení hladiny krevního cukru. Diabetik by si pak měl dávat pozor při dávkování inzulinu. Dávky by měl upravovat podle vydané pohybové aktivity [31].

U diabetiků s typem I je to poněkud složitější. Především je problém v tom,

že jejich tělo není schopno vyrobit dostatek inzulinu. Z tohoto důvodu by si měli každou sportovní aktivitu naplánovat, aby u nich nedošlo k akutním zdravotním komplikacím jako je *hypoglykémie* nebo *hyperglykémie*.

2.1.2 Teplota těla

Běžná teplota lidského těla se pohybuje v rozmezí 36 až 37°C [30]. Teplota může indikovat zdravotní problém, pokud se dostane mimo uvedenou mez. Při velkých ztrátách tepla lidského těla dochází k podchlazení (při teplotě menší než 36 °C). Zvláště u diabetiků je tento stav nebezpečný. Pokud člověk s diabetem trpí špatným prokrvením končetin, může vzniknout poškození nohou [13]. Toto poškození se u diabetiků velmi špatně hojí.

V případě zvýšené teploty (teplota nad 37 °C) má lidské tělo v sobě více tepla, než jeho organismus dokáže zpracovat [30]. K tomuto stavu může dojít například přehřátím z prostředí, ve kterém se člověk nachází. Může k němu dojít i na základě vnitřního podnětu lidského těla, a to zejména kvůli různým nemocem. Při zvýšené teplotě se tělo snaží ochladit jedním z následujících způsobů [13]:

- Mechanismus pocení
- Roztažení cév
- Zvýšená intenzita dýchání

Při nadměrném pocení se tělo rychleji odvodňuje, zvyšuje se hladina cukru v krvi a člověk má stále žízeň. U osob s dlouhodobým onemocněním diabetem může docházet k problémům s cévami. Při špatném odvodu tepla z organismu může docházet k infarktu nebo k cévní mozkové příhodě [30].

2.1.3 Okolní teplota

Teplota lidského těla souvisí s vnější teplotou. V těle jsou mechanismy reagující na změny teploty okolí. Pokud je tělo vystaveno chladu aktivují se mechanismy jako svalový třes. Zároveň dochází ke stažení cév. Při opačném jevu, pokud je tělo vystaveno vyšší teplotě dochází k pocení (viz sekce 2.1.2)

a intenzivnějšímu dýchání. Při vyšších teplotách dochází ke snížení chuti na jídlo, apatii a zvýšení nečinnosti [13].

V ideálním případě není nutné tyto mechanismy aktivovat. V případě, že to tak není, je potřeba je zohlednit. Příjem a výdej energie či hladina cukru v krvi totiž na teplotě okolí zcela jistě závisí.

Teplota prostředí negativně ovlivňuje úmrtnost lidí s diabetem [1]. Studie prezentuje závěr, že je zvýšené riziko úmrtí při vyšších teplotách okolí. Nicméně v tropickém prostředí může mít negativní vliv i nízká okolní teplota. Nízká teplota může mít pozitivní účinek na diabetiky s typem II. Při nízkých teplotách u těchto pacientů dochází k lepší regulaci glykémie, čehož je dosaženo zvýšenou citlivostí k inzulinu [1].

2.1.4 Krevní tlak

Jedním z častých doprovodných symptomů diabetu typu I a II je zvýšený krevní tlak [23].

Zvýšený krevní tlak zvyšuje riziko dalších zdravotních komplikací. Zdravotní komplikace u zvýšeného tlaku mohou vést až ke smrti. U diabetiků trpících vysokým tlakem má od určité hodnoty smysl snižovat krevní tlak. Neplatí to však pro všechny, je to velice individuální pro každého pacienta [23].

Hodnota krevního tlaku by měla být pravidelně sledována praktickým lékařem pacienta [17].

2.1.5 Stres

Stres je složité nadefinovat a obecně vůbec měřit. Jedna z definic říká, že stres je definován jako: *"Je to funkce těla reagující na mimořádné stavy jako jsou ohrožení, psychická nebo fyzická bariéra"*[32].

Pro diabetiky je stres další položka, u které by měli zpozornět. Jak je všeobecně známo, stres má vliv na metabolismus. Z důvodu působení stresu na člověka dochází ke změně rychlosti metabolismu [27]. Z tohoto důvodu by se měli diabetici snažit co nejvíce vyvarovat stresu a zároveň hledat metody, jak stresu předcházet. Pomalejší metabolismus znamená, že dochází k jiné regulaci inzulinu a glukózy u pacientů [27]. Pokud člověk trpí dlouhodobým

stresem, je potřeba tomu uzpůsobit i dávky inzulinu.

Stres je možné měřit nepřímo například pomocí tepové frekvence nebo pomocí měření změny povrchového (kůže) napětí [27].

2.2 Přenos dat

Jedním z hlavních cílů práce je přenášet z měřicího zařízení naměřená data. Přenos dat bude realizován pomocí některé z dostupných bezdrátových technologií. Níže jsou vypsány nejpoužívanější dostupné technologie pro bezdrátový přenos dat. Popsány jsou jejich klady a zápory.

Závěr sekce bude obsahovat zhodnocení a výběr vhodné technologie pro přenos dat.

2.2.1 IEEE 802.11

IEEE 802.11 je standard pro přenášení dat po bezdrátových sítích. IEEE 802.11 je spíše znám pod svou zkratkou `Wi-Fi`. Pro přenos dat se využívá většinou bezlicenčních frekvenčních pásem 2.4, 5 a 60 GHz. Samotný IEEE 802.11 se pak dělí na několik dalších standardů [15]. Tyto standardy se pak liší především tím, jaká využívají frekvenční pásma, maximální přenosovou rychlost, zabezpečení a další vlastnosti. Především v rychlostech je pak mezi jednotlivými standardy obrovský rozdíl. Na nejstarším standardu 802.11a je maximální teoretická přenosová rychlost v řádech desítek megabitů za sekundu, zatímco na nejnovějším se tato rychlost pohybuje v řádech gigabitů za sekundu. Při použití některého ze standardů z rodiny 802.11 je možno přenášet data v řádech desítek metrů. Samozřejmě s přihlédnutím na typ použité antény, frekvenci, vysílacímu výkonu a překážkám v trase k přijímači.

Na standardu IEEE 802.11 je dobře řešené zabezpečení přenosu dat. Je to z důvodu velkého využívání standardu při přenosu dat na bezdrátových sítích. Pro standard existuje několik metod šifrování dat [33], které se neustále v průběhu let vyvíjí, jak starší metody zastarávají a stávají se zranitelnými.

Výhody a nevýhody

Velkou výhodou je především rozšířenost této technologie, kdy existuje nepřeberné množství zařízení, které ji podporují. Velká přenosová rychlost na nejnovějších standardech je také jednou z výhod. Nicméně vzhledem k datům, která budou přenášena, nebude vysoká přenosová rychlost potřeba. Vzhledem k povaze dat, která budou přenášena, je velkou výhodou zabezpečení, které standard podporuje.

Při použití této technologie k přenosu dat, bude pravděpodobně vyšší spotřeba elektrické energie [3]. To bude mít i přímý vliv na výdrž baterie, která bude napájet zařízení. Další nevýhodou může být silné zarušení frekvenčních pásem v hustě zastavěných městských částech.

2.2.2 IEEE 802.15.4

IEEE 802.15.4 specifikuje nízkoúrovňovou komunikaci v nižších vrstvách v sítích typu `PAN`. Ze standardu pak vychází protokoly jako jsou `Zigbee`, `WirelessHART`, `MiWi` a další, které specifikují komunikaci na vyšších vrstvách.

Komunikace probíhá na bezlicenčních frekvenčních pásmech pro daný region kolem 800MHz nebo na 2.4GHz. Přenosová rychlost se pohybuje v řádech desítek kilobitů za sekundu. Vzdálenost, na jakou může probíhat přenos, je v jednotkách metrů [21]

Na standardu je možné šifrování dat pomocí šifrovací metody AES [10].

Výhody a nevýhody

Technologie neposkytuje závratné přenosové rychlosti dat. Tato rychlost je přesto dostačující pro účel, který má zařízení plnit, a to přenášet data posbíraná ze senzorů. I spotřeba elektrické energie je oproti konkurentům nízká [3]. Další výhodou je možnost šifrování přenášených dat.

Tato technologie se téměř nevyskytuje na mobilních zařízeních. V tom případě by k použití této technologie bylo potřeba pro větší dostupnost ještě nějakého třetího zařízení. To by pak poskytovalo data skrze rozšířenější technologii jako je např. *Wi-Fi* (viz sekce 2.2.1).

2.2.3 Bluetooth

Bluetooth je standard popisující komunikaci na bezdrátových technologiích. Pro svou komunikaci používá bezlicenční frekvenční pásmo 2.4GHz. Na zmíněném frekvenčním pásmu je pak pro Evropu 79 radiových kanálů o šířce 1 MHz. Rozprostření pásma je zajištěno technologií FHSS, kdy je provedeno 1600 přeskoků mezi všemi kanály během jedné vteřiny. Tento mechanismus má zvýšit odolnost vůči rušení. Přenosová rychlost se liší podle verze *Bluetooth*, kdy se rychlost v nejnovějších verzích pohybuje v desítkách megabitů za sekundu.

V Bluetooth je možné zabezpečení dat. Metody zabezpečení se liší v závislosti na verzi. V novějších verzích probíhá zabezpečení komunikace tak, že nejdříve probíhá párování zařízení, kdy se vyměnují sdílené tajné klíče. Klíče se používají v šifrovacích algoritmech pro zašifrování přenášených dat . Novější verze *Bluetooth* používají k šifrování dat algoritmus využívající metodu eliptických křivek [12].

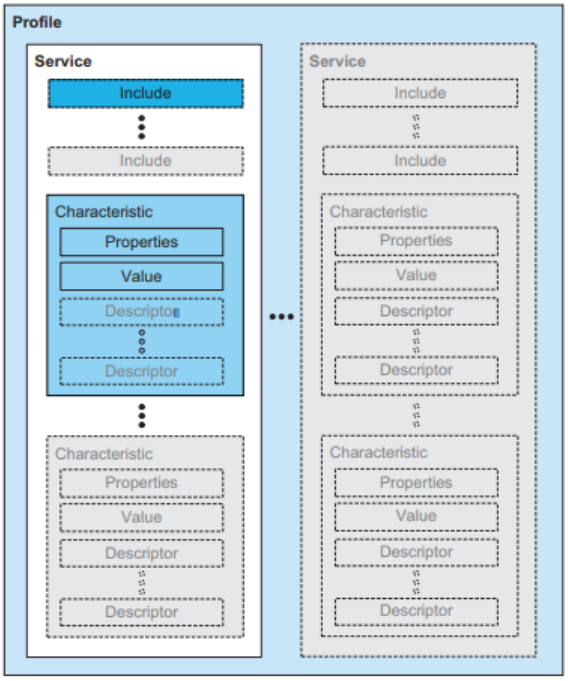
Bluetooth Low Energy

Bluetooth low energy, nebo také ve zkratce *BLE*, je technologie ve standardu *Bluetooth* zařazena od verze 4.0 a novější. Tato technologie se vyznačuje svou nízkou spotřebou energie. Nižší spotřeba energie je obětována na úkor vyšší propustnosti dat. V pásmu 2.4GHz využívá 40 kanálů o šírce 2MHz. Podobné je pak skákání mezi kanály, ale v technických detailech se liší, kvůli šetření energie.

Součástí BLE je sada profilů označující se zkratkou GATT. Ty definují jakým způsobem přenáší připojené BLE zařízení data. Data jsou v GATT hiearchicky členěna na *služby*, což jsou skupiny vzájemně souvisejících dat označené jako *charakteristiky*. Hiearchie GATT je dobře vidět na obrázku 2.1. Jelikož GATT popisuje pouze přenos dat z již připojeného zařízení, operuje nad ním ještě GAP. To je profil jednotného přístupu, který zajišťuje oznamování jaké služby a charakteristiky jsou dostupné.

GAP definuje čtyři role jakými může být provedeno připojení:

- **Broadcaster** - Zařízení pouze vysílá oznamovací pakety. V tomto režimu není možné spojení.



Obrázek 2.1: Hierarchie GATT [7]

- **Observer** - Zařízení pouze přijímá oznamovací pakety. V tomto režimu není možné připojení.
- **Central** - Zařízení aktivně nevysílá, ale čeká, až zařízení v roli *Peripheral* vyšle oznamovací paket. Když zachytí oznamovací paket, umožní navázání spojení se zařízení, které ho vyslalo.
- **Peripheral** - Zařízení vysílá oznamovací pakety, aby umožnilo spojení se zařízením v roli *Central*.

Pod GATT operuje protokol ATT. Ten zprostředkuje přenos dat. Data, ke kterým je umožněn přístup, jsou idenfikovaná jednoznačným identifikátorem UUID [20]. Každá datová jednotka může mít povolené některé z následujících datových přístupů:

- **Read request** - Požadavek na čtení dat z datového bloku.
- **Write request** - Požadavek na zápis dat z datového bloku.
- **Indication** - Přenos dat ze serveru na klienta, kdy data nejsou potvrzována.

- **Notification** - Přenos dat ze serveru na klienta, kdy data jsou potvrzována.

Aby server mohl komunikovat s klientem pomocí *Indication* nebo *Notification*, musí nejdřív klient tuto akci povolit.

V charakteristice jsou obsažené atributy. Ty říkají jakým způsobem je možné přistoupit k datům. Atributy určují jaké zprávy je možné přes charakteristiku přenést, zda je vyžadováno šifrování zpráv a další vlastnosti. V charakteristice je vždy obsažen hlavní atribut, a to je hodnota přenášená pomocí charakteristiky.

Výhody a nevýhody

Posuzování výhod a nevýhod bude zaměřeno především na *BLE*.

Důvod, proč bude posuzováno BLE a ne *Bluetooth* jako celek, je nízká spotřeba energie. Svou nízkou spotřebou energie [3] se stejně jako standart IEEE 802.15.4 (viz sekce 2.2.2) zařazuje mezi vhodné kandidáty pro použití v měřicím zařízení. Hlavní výhodou oproti standardům vycházejícím z IEEE 802.15.4 (viz sekce 2.2.2) je pak velká dostupnost na mobilních zařízeních, laptopech a stolních počítačích. Samozřejmostí je i možnost šifrování dat.

Co může být nevýhodou, je náchylnost k rušení i přes použití technologií, které by vůči němu měly zvyšovat odolnost. Důvodem je rozšířenost *Wi-Fi*, která vysílá na stejné frekvenci jako *Bluetooth*. Pak např. v bytových domech nebo nákupních centrech dochází kvůli velké hustotě *Wi-Fi* vysílačů i k rušení *Bluetooth*.

2.2.4 Mobilní síť

Mobilní síť je tvořena množstvím základových stanic tzv. *BTS* umístěných na vhodných geografických místech. Stanice se signálem vzájemně překrývají na různých frekvencích a tvoří malé buňky. Podle typu použitých frekvencí na buňce se odvíjí i vzdálenost, na kterou je schopné zařízení se k buňce připojit. Každá buňka pak funguje jako vysílač a přijímač pro zařízení, které se k ní připojí. Stanice jsou navzájem propojené přes komplexní systém,

který umožňuje přenos dat z jednoho zařízení na druhé. Zařízení je vždy spojené s buňkou, pokud je dostupný signál. Systém buněk si umí předávat zařízení mezi buňkami, které je pro ně výhodnější.

Nejrozšířenější a nejstarší je *GSM*, která se používá výhradně pro přenos hlasu a textových zpráv. Jsou tu ale i modernější technologie jako *GPRS*, *Edge*, *UMTS*, *HSPA*, *LTE* a další, které se také používají pro přenos hlasu, textových zpráv nebo dat. Na zmíněných technologiích je možné zabezpečení pomocí šifrování. Úroveň zabezpečení se liší podle použití některé ze zmíněných technologiích používaných v mobilních sítích. Například u nejstarší technologie *GSM*, která se používala ještě pár let zpátky, bylo zabezpečení dat velmi špatné [16]. Oproti tomu v současnosti nejrozšířenější *LTE* už poskytuje dostatečné zabezpečení dat.

NB-IoT

Tato technologie vychází z technologie *LTE* a je určená výhradně k přenosu dat. Technologie je navržená tak, aby byly nízké nároky na spotřebu energie při přenosu dat. Dá se využít na nízkopříkonových zařízeních, ale je potřeba zařízení opatřit SIM kartou a platit za přenesená data.

Výhody a nevýhody

Nespornou výhodou použití mobilní sítě je, že signál je dostupný téměř všude. Za použití *GSM* modulu by pak zařízení navrhované v této práci mohlo přenášet data téměř odkudkoliv. V sítích se používají licencovaná frekvenční pásma, u kterých je menší pravděpodobnost k výskytu rušení. V případě použití nejnovějších standardů jako je *LTE*, je zabezpečení na dobré úrovni [16].

Nevýhoda může být, že připojení k síti je zpoplatněno. Platí se za objem přenesených dat. Spotřeba energie při přenosu dat může být vyšší oproti jiným technologiím [18]. Šifrování je možné, ale aby bylo proveditelné, potřebuje každé zařízení používající tuto technologii SIM kartu.

2.2.5 Shrnutí

Každou z výše uvedených technologií by bylo možno použít pro měřicí zařízení vytvářené v rámci této práce. Je ale potřeba myslet na požadavky, které by měla technologie pro komunikaci splňovat. Komunikace by měla zajišťovat přenos hodnot získaných ze senzorů připojených k měřicímu zařízení tvořenému v rámci této práce. Očekává se získávání dat ze senzorů v řádu desítek bajtů každou sekundu. Z toho důvodu bude dostačující přenosová rychlost v řádech Kb/s, aby nedocházelo k vysoké odezvě.

Dále je potřeba, aby navrhované zařízení šlo připojit na některé z běžných zařízení jako je mobilní telefon, laptop, stolní počítač, a další. Je tedy třeba, aby technologie byla dostatečně rozšířená. Důležitým hlediskem pro navrhované zařízení bude spotřeba el. energie, protože bude zařízení napájeno z baterie. Je žádoucí, aby technologie měla co nejmenší nároky na spotřebu el. energie, aby navrhované zařízení vydrželo co nejdéle. Jednou z méně důležitých vlastností je bezplatnost přenosu dat na dané technologii. Bezplatnost se netýká mobilních sítí, kde se předplácí maximální objem přenesených dat za jeden měsíc. Poslední vlastností je náchylnost k rušení. Rušení může být jedním z důležitých faktorů ovlivňující, přenosovou rychlost a spolehlivost přenosu. Rušení se týká především technologií využívajících bezlicenčních pásem na frekvenci 2.4GHz. Zabezpečení je neméně důležitá vlastnost a to především v případě, kdy se mají přenášet medicínská data. Vezmou-li se v potaz pouze nejnovější verze všech zmiňovaných technologií, tak všechny poskytují dostatečnou úroveň zabezpečení.

Vlastnosti výše popisovaných technologií jsou shrnuty v následující tabulce 2.1. Pro položky spotřeba, rušení, rozšířenost je použita stupnice *nízká*, *střední* a *velká*. Pro položku spotřeby a rušení je *nízká* vhonější a naopak velká *horší*. U rozšířenosti je *velká* lepší a *nízká* špatná. U položky rychlost mohou být hodnoty b/s až Gb/s, kdy jsou lepší hodnoty Gb/s. V poslední řadě položka bezplatnost, kde jsou dvě možnosti buď *ano* nebo *ne*. Je samozřejmě lepší, když se dá využít technologie bezplatně.

Na základě předchozích sekcí a tabulky 2.1 se může zdát optimální využít technologii 802.15.4. Tato technologie splňuje všechna požadovaná kritéria, kdy má nízkou spotřebu energie a dostatečnou rychlost. Kromě toho, díky využívání frekvečních pásem na frevenci 800MHz, je zde vyšší pravděpodobnost bezproblémového chodu, protože toto pásmo není tolik zarušené. Je ale velký problém, že není rozšířena na zařízeních typu mobilní telefon,

laptop, stolní počítač a další. Pro využití sítě by bylo nutné mít vstupní bod, který by dokázal komunikovat, jak s technologií IEEE 802.15.4, tak s nějakou běžně dostupnou např. Ethernet nebo IEEE 802.11.

Mobilní síť je výhodná v tom, že má téměř 100% pokrytí signálem. Připojit se k navrhovanému zařízení na tomto typu sítě by pak šlo teoreticky odkudkoliv z Internetu. Technologie je nevýhodná v tom, že je přenos dat zpoplatněn a spotřeba el. energie je také velká oproti ostatním zmíněným [18].

Nejvhodnější volba je technologie BLE jako dílčí standard technologie *Bluetooth*. Nabízí stejně jako IEEE 802.15.4 nízkou spotřebu energie. Rychlost přenosu dat je naprosto dostačující. Rozšíření na zařízeních, jako je mobilní telefon nebo laptop, je velké [8] .Technologie má tu nevýhodu, že využívá frekvenční pásmo, které je ve městech silně zarušené. Frekvenční skoky [24] používané na této technologii dokáží vliv rušení zmenšit.

	spotřeba	rychlost	rušení	rozšířenost	bezplatnost	zabezpečení
IEEE 802.11	velká	až Gb/s	střední až velká	velká	ano	ano
IEEE 802.15.4	nízká	Kb/s	nízká	nízká	ano	ano
Bluetooth	střední	Mb/s	střední	velká	ano	ano
BLE	nízká	Mb/s	střední	velká	ano	ano
Mobilní síť	velká	Mb/s	nízká	střední	ne	ano

Tabulka 2.1: Tabulka srovnání bezdrátových technologií

2.3 Platforma měřicího zařízení

Pro sběr výše zmíněných veličin (viz sekce 2.1) je potřeba vybrat vhodné zařízení. Po takovém zařízení bude vyžadováno, aby zvládlo obsluhovat několik senzorů zároveň. Mělo by být možné senzory přidávat, tak i odebírat podle potřeby. Zařízení by mělo umožňovat komunikaci pomocí technologie *BLE*, protože tato technologie vyšla v předchozí sekci jako nejlepší pro přenos dat z měřicího zařízení.

V analýze budou rozebrány čtyři z nejrozšířenějších platforem hodících se pro sběrné zařízení. U platforem se bude brát v potaz výkonnost hardware, energetická úspornost, podpora vývoje ze strany výrobce, komunitní podpora a podpůrné nástroje pro vývoj. Posledním sledovaným aspektem budou dostupná komunikační rozhraní např. UART, I2C, SPI.

Ke každé platformě bude uveden jeden zástupce poskytované vývojové desky, který se svými parametry hodí nejvíce pro tuto práci a zároveň je dosažitelný na českém trhu.

2.3.1 Arduino

Tato platforma je dobrou volbou pro začátečníky, kteří začínají programovat na mikrokontrolery. Na Internetu je velké množství dostupných návodů, jak pro začátečníky, tak pokročilé. *Arduino* má vlastní IDE pro vývoj kódu pro mikrokontrolery a vlastní programovací jazyk *Wiring* [4] postavený nad C/C++. Jazyk je navržen tak, že i ti kteří neprogramují, se snadno naučí vytvářet vlastní kód. Na většinu běžně prodávaných senzorů se dá pro tuto platformu nalézt knihovna, což velice usnadňuje vývoj kódu.

Když se podíváme, jaký hardware je běžně dostupný na platformu *Arduino*, máme na výběr ze tří desek. Arduino UNO, Arduino NANO a Arduino MEGA. Tyto desky jsou osazeny nepříliš výkonnými 8bitovými mikroprocesory ATmega328. Mikroprocesor nabízí taktovací frekvenci pouze 20MHz. K dispozici je jeden 10bitový ADC převodník a rozhraní pro UART, I2C a SPI. Pro řídící kód je pak vyhrazena 32KB flash a 2KB SRAM.

2.3.2 Espressif ESP

Tato platforma je na vývoj kódu už poněkud náročná. Výrobce dodává zásuvný modul do IDE Eclipse. Kód pro tuto platformu se vyvíjí v jazyce C/C++. Výrobce má k dispozici rozsáhlou dokumentaci API. U této platformy není tak rozsáhlá uživatelská komunita. V případě problému je vývojář odkázaný většinou pouze na dostupnou dokumentaci. Je zde i jednodušší varianta, jak vyvíjet kód na tuto platformu a to pomocí Arduino IDE. Jde to pomocí zásuvného modulu do Arduino IDE. Vývoj software pak probíhá podobně jako na platformu *Arduino*.

Deska od platformy *ESP*, která by byla pro práci zajímavá, je IoT ESP-WROOM-32 2.4GHz Dual-Mode WiFi+Bluetooth rev.1, CP2102. Tato deska je osazena mikrokontrolerem ESP-WROOM-32. Tento mikrokontroler se vyznačuje svým velkým výkonem a zároveň integrovanou *Wi-Fi* s *Bluetooth*. Mikrokontroler obsahuje dvě jádra, která je možné nataktovat až na 240MHz. K tomu poskytuje 4 MB flash paměti a 520KB SRAM. Co se týká dostupných rozhraní, obsahuje 3x UART, 3x SPI, 2x I2S, 2x 12bitový ADC převodních s 12 kanály, 2x 8bitový DAC převodník a 2x I2C rozhraní.

2.3.3 Texas Instruments

Výrobce má k dispozici materiály s návody na vývoj kódu, ale jsou rozhodně nedostačující pro jednoduchý začátek s vývojem na platformu *Texas Instruments*. Kód se vyvíjí v jazyce C/C++. Výrobce poskytuje software *Code Composer Studio*, což je vývojové IDE pro platformu *Texas Instruments*.

Oproti ostatním zmiňovaným platformám je těžší získat vhodnou desku na českém trhu. Desky se dají získat pouze ve dvou specializovaných obchodech, přičemž pouze jeden má pobočku v ČR. Jako vhodný zástupce byla vybrána deska LaunchXL CC2640R2. Deska je osazena 32bitovým mikrokontrolerem ARM Cortex-M3. Ten má v sobě zabudovaný BLE modul. Mikrokontroler dosahuje taktovací frekvence 48Mhz s flash pamětí 128KB a 20KB SRAM. Mikrokontroler je vybaven 32 kanálovým DMA řadičem. Dále jsou k dispozici 1x I2C rozhraní, 8x UART a 1x 12bitový ADC převodník.

2.3.4 STM32

Platforma *STM32* je v porovnání s Arduinem už složitější pro vývoj vlastního kódu. Firma dává vývojářům k dispozici vlastní IDE. Jedná se o IDE Eclipse rozšířené o doplňky pro vývoj na platformě *STM32*. V tom je možné si vybrat mikrokontroler, pro který bude prováděn vývoj, a umožňuje předgenerovat si základní strukturu kódu pro další vývoj. Platforma má rozsáhlou komunitu. Výrobce má obsáhlý manuál s kompletní množinou informací pro vývoj, ale i videonávody pro ukázku vývoje nejběžnějších postupů. Vyvíjí se na tuto platformu v jazyce C/C++ a jako základ jsou knihovny HAL zprostředkující komunikaci s hardwarovým vybavením desky. Díky tomu je možné měnit cílový mikrokontroler bez nutnosti měnit kód. Pro jednodušší vývoj je možné použít modul do Arduino IDE.

Tato platforma poskytuje velké množství desek s různými typy mikrokontrolerů. Ne všechny desky jsou běžně k dostání v Evropě a může být na ně velká čekací doba. Z toho důvodu se v další části zaměříme na jednu z nejdostupnějších desek a to STM32F103C8T6. Tato deska je osazena 32bitovým mikrokontrolerem STM32F103C8. Mikrokontroler nabízí výkon 72MHz s 64KB místem ve flash paměti a 20KB místa na SRAM. Jsou dostupné dva 12bitové ADC převodníky, 9 komunikačních rozhraní (2x I2C, 3x UART, 2x SPI, CAN a USP ve verzi 2.0). Kromě toho je k dispozici 7 kanálový DMA kontrolér umožňující přenos dat mezi rozhraními a pamětí bez potřeby využití procesoru.

2.3.5 Shrnutí

Arduino nenabízí příliš vykonné desky dostupné na tuzemské trhu, jak je vidět v tabulce 2.2. *Arduino* nabízí i výkonnější desky, ale buď jsou těžko dostupné nebo jsou oproti ostatním drahé. Při srovnání s ostatními deskami dojdeme k závěru, že co se výkonu procesoru, flash paměti a SRAM paměti týká, jsou na tom nejlépe desky od *ESP* a *TI*. Hodnoty spotřeby elektrické energie mikrokontrolerů v různých konfiguracích bez připojených periferií jsou srovnatelné, a je možné je najít v datových listech [28] [2] [26] [11]. Vrátíme-li se k samotnému porovnání výkonu, akceptovatelné platformy pro tuto práci jsou *ESP*, *TI* a *STM32*.

U navrhovaného zařízení bude potřeba, aby bylo dobře rozšiřitelné. Dosažitelnost tohoto kritéria je přehledně zanesena do tabulky 2.3. Výslednou

	taktovací frekvence procesoru	flash/SRAM
Arduino	8-bit 20MHz	32KB/2KB
ESP	32-bit 240MHz	4MB/520KB
TI	32-bit 48MHz	128KB/20KB
STM32	32-bit 72MHz	64KB/20KB

Tabulka 2.2: Srovnání základních parametrů desek

spotřebu desky může ovlivnit to, zda je nebo není integrován DMA řadič [6]. Je to dáno tím, že použitím DMA řadiče předejdeme zpracování dat procesorem. ADC převodníky se hodí pro měření napětí na pinech, které může mít mnohá využití (např. měření teploty). V tabulce je vidět že *Arduino* má pouze 10bitový ADC převodník oproti 12bitovým na ostatních mikrokontrolerech. Rozdíl mezi nimi je v přesnosti naměřených hodnot napětí na pinech, kde operuje ADC. Pokud je na senzoru použito rozhraní I2C, je často dostupné i rozhraní SPI. To, že není žádné rozhraní na desce od *TI*, nemusí být nevýhodou. Tato dvě rozhraní se používají pro komunikaci s různými typy senzorů (akcelerometry, měřiče vlhkosti, měřiče vzdálenosti, měřiče tlaku a spoustu dalších). Rozhraní I2C má tu vlastnost, že na jedno rozhraní může být připojeno až 128 senzorů. Díky tomu může stačit i jediné na mikrokontroleru. Posledním důležitým rozhraním je UART. Toto rozhraní se hodně používá na různých senzorech a také se využívá při vývoji pro výpis ladících zpráv. Pokud mikrokontroler nemá *BLE* zabudován v sobě, toto rozhraní se používá pro komunikaci s *BLE* modulem. Z celkového srovnání hardwarových desek vyjdou přijatelně desky od *ESP*, *TI* a *STM32*. Platforma *Arduino* neobsahuje DMA řadič a také ADC převodník je méně přesný než u ostatních platforem.

	DMA	ADC	I2C	SPI	UART
Arduino	ne	1x 10-bit	2x	2x	1x
ESP	ano	2x 12-bit	2x	2x	3x
TI	ano	1x 12-bit	1x	ne	1x
STM32	ano	2x 12-bit	2x	2x	3x

Tabulka 2.3: Srovnání vybavení desek

Poslední část se týká srovnání samotného vývoje na danou platformu. V této části byla porovnávána tři kritéria, jak je vidět v tabulce 2.4, a to dostupnost

hardware, úroveň podpory výrobce a komunitní podpora. Nejlépe, jak je vidět v tabulce 2.4, vychází *Arduino*. Deska od *Arduina* je dobře dostupná. Na Internetu je množství návodů, knih a videonávodů, které pomáhají s vývojem. *Arduino* má velkou komunitu, takže lze rychle a efektivně vzniklé problémy řešit v rámci on-line diskusí. Z hlediska vývoje kódu na platformu jsou na tom ostatní tři výrobci hůře než *Arduino*. To lze odůvodnit například tím, že *Arduino* je přizpůsobeno uživatelům bez zkušeností s programováním a řada mechanismů a rozhraní je tak výrazně zjednodušena. Firma *TI* není příliš rozšířená na českém trhu. Jediným distributorem pro Českou Republiku je tuzemský obchod *Mouser*, od kterého se dají získat desky. Ani komunitní podpora není velká. To samé se týká dokumentace poskytnuté od firmy *TI*, která je celkem náročná pochopení. Trochu lépe je na tom platforma *ESP*, kde je lepší dostupnost desek na tuzemském trhu. Příjemná je dokumentace od výrobce *ESP*, kde se dá nalézt mnoho postupů a ukázek kódu. To samé se týká i komunitní podpory. Poslední platforma *STM32* nemá velké zastoupení desek dostupných v tuzemských obchodech. Deska od *STM32* (viz sekce 2.3.4) je však dostupná i v řadě tuzemských obchodů. Komunitní podpora je velmi rozsáhlá. Podpora od výrobce také není špatná. Jsou k dispozici jak psané návody, tak videonávody na stránkách výrobce.

S ohledem na všechna tři kritéria hodnocení vychází platforma *STM32* jako nejlepší varianta. Výkon desky od *STM32* (viz sekce 2.3.4) je dostačující a poskytuje potřebný počet rozhraní pro připojitelné senzory. Vzhledem k velikosti komunity a rozsáhlé podpoře od výrobce lze konstatovat, že vývoj na platformu *STM32* není tak snadný jako na *Arduino*, ale snazší než na *ESP* nebo *TI*.

	Dostupnost hardware	komunitní podpora	dostupnost návodů
Arduino	velká	velká	velká
ESP	střední	střední	střední
TI	malá	malá	střední
STM32	velká	velká	střední

Tabulka 2.4: Tabulka srovnání platforem

2.4 Senzory

Sekce je zaměřena na analýzu senzorů. Níže budou vypsány senzory schopné měřit veličiny vycházející z analýzy (viz sekce 2.1).

2.4.1 Akcelerometr s gyroskopem

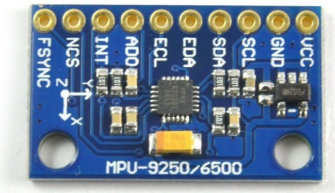
Akcelerometr je zařízení měřící hodnoty posuvného a rotačního zrychlení. Zrychlení udává informaci o tom, jak se mění rychlost tělesa v čase (nebo úhel v případě rotačního). Toto zařízení produkuje celkem šest hodnot. Tři hodnoty se týkají posuvného zrychlení v osách x, y a z. Zbylé tři pak rotačního zrychlení, taktéž v osách x, y a z.

Na základě těchto dat se může vypočítat, jakou aktivitu osoba vykonávala. Může se jednat i o klidovou aktivitu, kupříkladu spánek. To pak budou hodnoty získané z akcelometru velmi nízké a stálé. V opačném případě může jít o pohybovou aktivitu jako běh, cyklistika, rychlá chůze a jiné. V této fázi budou hodnoty naopak mnohem rozličnější a vyšší.

Samozřejmě měřicí zařízení nedokáže určit bez dalších senzorů, zda sledovaná osoba doopravdy nějakou aktivitu vykonává. Akcelerometr sice dokáže získat informaci o zrychlení, ale už nedokáže určit kontext získání této informace. Osoba, jejíž pohybová aktivita bude měřena akcelerometrem, se může přesouvat pomocí automobilu. Senzor pak nebude vracet hodnoty zrychlení osoby, ale automobilu.

Pomineme-li pohybovou aktivitu, jsou tu i neočekávané situace, které mohou u člověka s diabetem nastat. Může nastat např. ztráta vědomí. Při ztrátě vědomí může dojít k pádu. Zařízení se pak o pádu dozví díky kombinaci dat získaných z hodnot posuvného a rotačního zrychlení.

K získání informací o posuvném a rotačním zrychlení se dá využít například senzor MPU-9250 vyobrazený na obrázku 2.2 obsahující tříosý akcelerometr, gyroskop a magnetometr.



Obrázek 2.2: Senzor MPU-9250, Zdroj: `https://dratek.cz/arduino/1330-9dof-gyroskop-akcelerometr-magnetometr-mpu-9250-spi-iic-modul-pro-arduino.html`

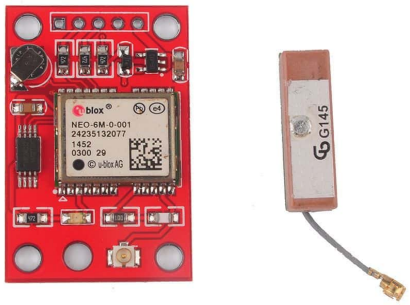
2.4.2 GPS modul

GPS je systém určený pro lokalizaci polohy. Kromě lokalizace se dá dobře využít i pro zjištění aktuální rychlosti. Ta se dá z GPS zařízení získat na základě výpočtu změny polohy v čase.

V kombinaci s akcelerometrem lze pak zpřesnit detekci aktivity, kdy dokáže lépe určit co sledovaná osoba asi dělá. Například bude-li se nacházet v lese a bude se pohybovat rychlostí např. 8-12 km/h, pravděpodobně běží. Naopak bude-li osoba na souřadnicích, kde se vyskytuje silniční komunikace a rychlost bude přes 50 km/h, pravděpodobně někam jede automobilem.

I v uvedených případech není jisté, že se jedná o pohybovou aktivitu. Proto je potřeba kombinace dalších senzorů. Vhodná by byla kombinace se senzorem tepové frekvence a senzorem pro snímání teploty těla.

Lokalizaci může provádět například senzor NEO-6M, který umí získávat GPS souřadnice pomocí družic. Podoba senzoru je vyobrazena na obrázku 2.3.

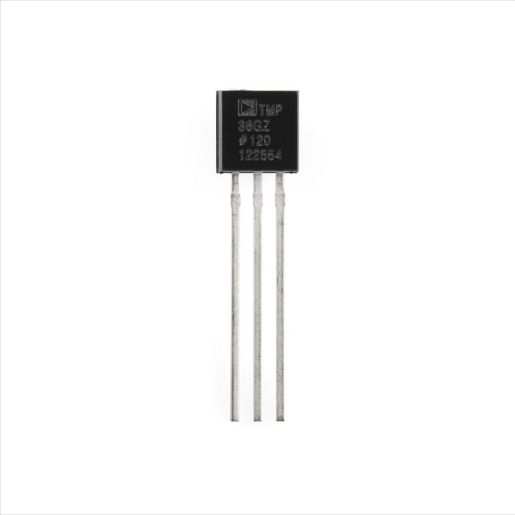


Obrázek 2.3: GPS modul, Zdroj: `https://www.elecdesignworks.com/shop/ublox-neo-6m-gps-module-red-mini-indoor-antenna/`

2.4.3 Měření teploty

Pro měření teploty, kterou bude získávat měřicí zařízení, případají v úvahu elektrické teplotní senzory. Těchto senzorů může být několik typů. Podstatné je, že tyto teploměry měří teplotu svého prostředí. To mění elektrické vlastnosti (např. úroveň napětí) teploměrů, ze kterých se dá zjistit hodnota teploty.

Jako senzor měření teploty se dá například využít elektronickou součástku obsahující diodu. Při změnách teploty se na diodě mění výstupní hodnota napětí z diody. Toto napětí se pohybuje v předem stanovených intervalech. Při znalosti výstupního napětí stačí tuto hodnotu zadat do vzorce a určit výslednou teplotu. Příkladem tohoto typu senzoru může být senzor TMP36 vyobrazený na obrázku 2.4, který obsahuje thermo-diodu.



Obrázek 2.4: Senzor teploty TMP36. Zdroj: `https://www.robotshop.com/en/temperature-sensor-tmp36.html`

2.4.4 Měření srdečního tepu

Pro měření srdečního tepu lze použít senzor, který využívá optický snímač využívající PPG metodu. Senzor obsahuje optický snímač v podobě svítivé (LED) a světlocitlivé diody (senzor). Dioda prosvicuje kůži a senzor pak snímá množství odraženého světla. Množství odraženého světla je úměrné změnám objemu krve [9]. Senzory používají nejčastěji infračervenou diodu (IR-LED) pro měření průtoku krve. Je to z toho důvodu, že infračervené světlo se dokáže dostat hlouběji až k částem lidského těla, kde je krev více koncentrovaná (např. svaly) [9]. Světlocitlivé senzory získávají intenzitu odraženého infračerveného světla. Z tohoto pozorování senzor získává informace o průtoku krve. Při pozorování průtoku krve v čase lze vypočítat hodnotu srdečního tepu.

Na obrázku 2.5 je příklad senzoru využívající metody PPG. Takový senzor může být přiložen na prstu nebo na zápěstí, kde dokáže získávat hodnoty o průtoku krve.

Obrázek 2.5: Senzor MAX30100, Zdroj: `http://www.alselectro.com/max30100-pulse-oximeter-spo2-and-heart-rate-sensor-module.html`

3 Návrh

V této kapitole bude rozebrán návrh měřicího zařízení, protokolu pro přenos dat z měřicího zařízení, aplikace pro sběrné zařízení, filtr pro software SmartCGMS a protokol pro přenos dat do SmartCGMS.

3.1 Měřicí zařízení

K měřicímu zařízení budou připojeny čtyři senzory. Bude se jednat o dva teploměry, GPS a akcelerometr s gyroskopem. Tyto senzory budou pro potřeby této práce připojeny k měřicímu zařízení pomocí vodičů a pájivého pole. Měřicí zařízení nebude potřeba připojovat do elektrické sítě. Bude opatřeno dobíjecí baterií a vypínačem. Připojené senzory budou ve firmware reprezentovány jako moduly. Za modul se bude považovat logický celek reprezentující naměřené a zpracované veličiny. Data z měřicího zařízení se budou získávat pomocí navrženého protokolu (viz sekce 3.2) pomocí technologie BLE. Protokolem se budou provádět i nastavení měřicího zařízení.

3.1.1 Firmware

Firmware bude navržen tak, aby bylo možné senzory jednoduše přidat, ale i odebrat. To znamená že se vytvoří jádro, ve kterém se registrují moduly představující abstraktní skupiny veličin získávaných ze senzorů. Toho bude docíleno pomocí jednotného rozhraní, kdy každý modul bude mít nadefinovanou množinu funkcí, které bude poskytovat. Pro správnou funkci modulů se pak jen doimplementuje kód pro získávání a zpracování dat. V jádře se zaregistrují ty funkce, které budou vykonávat funkčnost jednotného rozhraní.

3.1.2 Detektor pohybu

Pro tento senzor budou vytvořeny tři moduly operující se senzorem. Moduly budou využívat pro komunikaci se senzorem rozhraní I2C.

Akcelerometr

První z modulů bude obstarávat akcelerometr. Ze senzoru z části akcelerometru se budou získávat tři hodnoty. Získané hodnoty jsou pouze nezpracovaná data. Podle nastavení je možné získávat hodnoty zrychlení v rozsahu $2G$, $4G$, $8G$ a $16G$. U modulu se bude nastavovat rozsah na $2G$.

Akcelerometr bude fungovat tak, že bude operovat vždy v krátkých intervalech, po které bude získávat hodnoty a poté bude uspán. Pomocí uspávání akcelerometru se docílí menší spotřeby energie, což povede k delší výdrži baterie.

Gyroskop

Stejně jako akcelerometr i gyroskop bude získávat tři hodnoty ze senzoru. Tyto hodnoty představují úhlové zrychlení na ose. U gyroskopu je taktéž možnost nastavení citlivosti. K dispozici jsou čtyři úrovně citlivosti, a to $250deg/s$, $500deg/s$, $1000deg/s$, $2000deg/s$. V rámci této práce bude využita nejnižší a zároveň i nepřesnější úroveň. Sám o sobě gyroskop vrací hodnotu, která se bude převádět na úhlovou rychlost.

Intenzita pohybu

Data z akcelerometru není možné přenášet v rozumném čase na měřicí zařízení za použití technologie BLE. Jde o to, že aby šlo určit alespoň přibližně intenzitu pohybu, je třeba získat několik jednotek dat během krátkého časové intervalu. Z tohoto důvodu bude v měřicím zařízení modul, který toto bude vykonávat a pošle pouze výslednou hodnotu intenzity pohybu.

Modul bude brát jako podklad celkový vektor zrychlení z hodnot získaných z akcelerometru. Podle velikosti tohoto vektoru se pak určí, jak velká intenzita pohybu byla prováděna. Intenzita se bude měřit na základě deseti

vzorků z akcelerometru. Vzorky budou představovat rozdíly velikosti vektoru celkového zrychlení mezi aktuální naměřenou hodnotou a předchozí naměřenou hodnotou. Ze vzorků se udělá průměr a tento průměr bude odpovídat intenzitě pohybu podle definované tabulky 3.1. Čím vyšší je stupeň intenzity v tabulce 3.1 tím vyšší je pohybová aktivita.

Stupeň intenzity	rozsah zrychlení v mG
1	[0,50)
2	[50,500)
3	[500,1000)
4	[1000,2000)
5	[2000,∞)

Tabulka 3.1: Tabulka intenzity pohybu

Navržený postup je pouze jednoduchá metoda pro měření intezity pohybu. V práci dále není popisovaná přesnost této metody.

3.1.3 Měření teploty

Jak vyplynulo z analýzy (viz sekce 2.1), bude se měřit okolní teplota a teplota těla. K měřicímu zařízení budou připojeny dva tyto senzory pro měření zmíněných teplot. Data ze senzorů se budou získávat pomocí ADC převodníku. To je prostředek pro změření úrovně napětí na určitém pinu. STM32F103C8 je opatřeno dvěma ADC převodníky. Pro získávání hodnot bude použit pouze jeden převodník s tím, že pro každý modul bude určen ADC kanál. Pro dosažení co největší přesnosti se bude za jedno měření získávat sto jednotek dat, které se budou průměrovat. Teploměr není potřeba vypínat, protože se jedná pouze o pasivní elektronickou součástku. Nějvětší spotřeba energie je generována ADC převodníkem.

Jeden z teploměrů se použije pro měření teploty lidského těla. Teplota se bude měřit ze hřbetu ruky. Druhý teploměr bude použit pro měření okolní teploty.

3.1.4 Lokalizace

Tímto senzorem budou získávány zeměpisné souřadnice. Senzor má vlastní komunikační protokol. Protokol je navržen tak, že po připojení začne periodicky posílat data. Na měřicím zařízení pak je, aby začalo „poslouchat“ a tato data přijímalo. Data se souřadnicemi se začnou posílat ve chvíli, kdy senzor nalezne dostatečný počet satelitů.

Interval mezi zasláním zprávy se souřadnicemi ze senzoru je přibližně 1 s. Interval se dá snížit nastavením vyšší frekvence zpracování dat na senzoru pomocí protokolu. V rámci této práce se frekvence nebude zvyšovat, protože by měla nežádoucí efekt na zvýšení spotřeby elektrické energie.

Senzor by bylo vhodné uspávat pro větší úsporu elektrické energie. To nebylo realizováno, protože výrobce nepopisuje ve svých manuálech, jak to provést. Dalším důvodem by mohla být dlouhá doba k určení aktuálních souřadnic.

3.1.5 Přenos dat

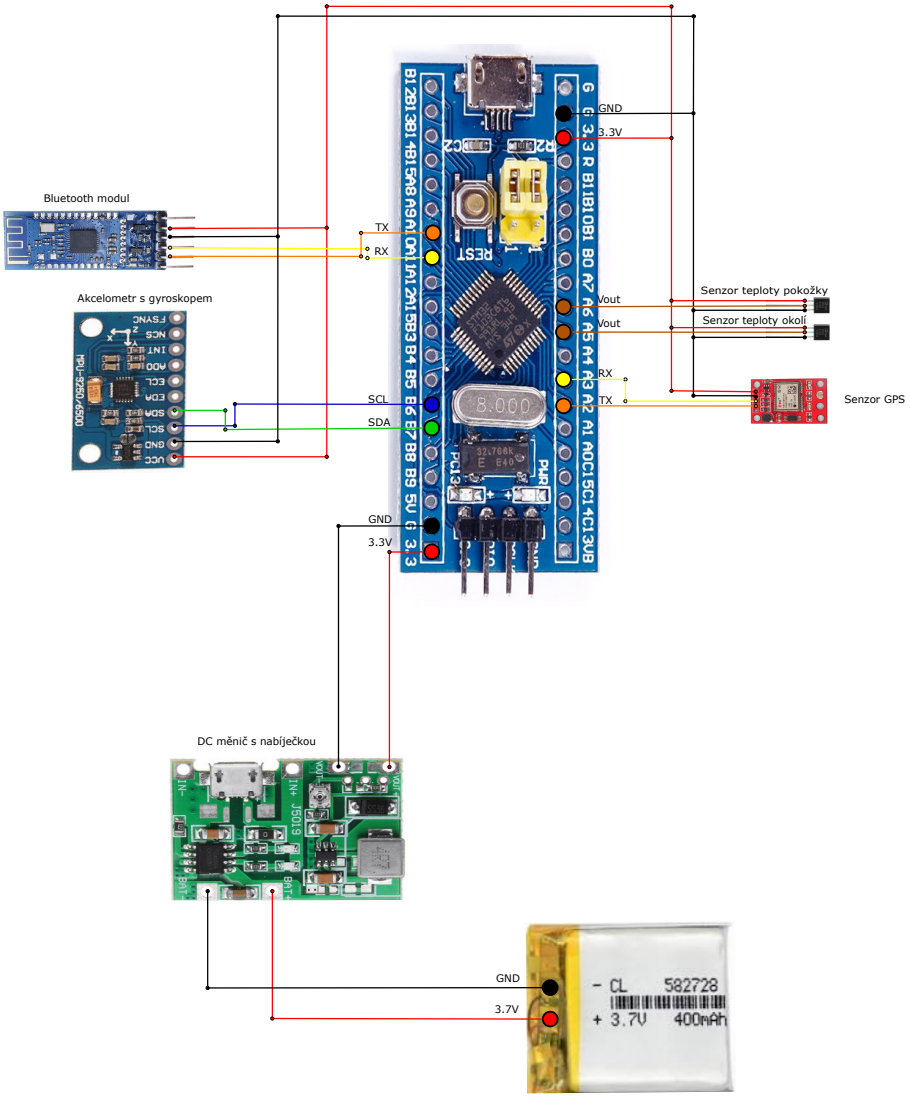
Pro přenos dat bude k zařízení připojen modul umožňující komunikaci pomocí BLE. Sběrné zařízení, které se bude k měřicímu zařízení pomocí tohoto

modulu připojovat, bude mít možnost zapnout si notifikaci na charakteristice. Notifikace bude dávat sběrnému zařízení vědět, že jsou připravena data pro čtení. To sběrnému zařízení umožní data číst hned, jak budou dostupná.

3.1.6 Schéma zapojení

Na obrázku 3.1 je návrh zapojení měřicího zařízení se všemi výše uvedenými senzory. Jak bylo uvedeno, měřicí zařízení bude napájené baterií. Baterie nebude připojena na měřicí zařízení přímo, ale skrze nabíječku. Nabíječka má v sobě DC/DC měnič a tím se zajistí, že do měřicího zařízení půjde napětí 3.3V.

Barvy vodičů na obrázku 3.1 budou ve výsledném zapojení stejné. Červený vodič bude použit na 3.3V výstup. Černý vodič se použije na uzemnění $3.3V$ okruhu. Pro rozhraní UART je použit oranžový vodič pro TX výstup z desky a žlutý vodič pro RX výstup z desky. Na I2C připojení se použije modrý a zelený vodič, kdy zelený bude pro SDA a modrý pro SCL. Poslední použitou barvou bude hnědá pro měření napětí ADC převodníkem.



Obrázek 3.1: Schéma zapojení sběrného zařízení

3.2 Přenos dat do sběrného zařízení

Aby bylo možné získávat data z měřicího zařízení, je nutné vytvořit aplikační protokol pro přenos těchto dat. Protokol je postaven na principu výzva-odpověď. Jedním z důvodů je, že některé BLE moduly mohou mít pouze jednu charakteristiku pro zápis a čtení dat. Problém by nastal, kdyby zpráva dorazila na modul dřív, než přijde odpověď na předchozí zprávu. Dojde k tomu, že pokud zařízení ještě zpracovává data z předchozí zprávy, dojde k jejich přepisu. Přepsat se mohou buď právě čtená data nebo data která se připravují na odeslání.

V případě, že na výzvu odpověď nepřijde, zkusí se po vypršení časového limitu poslat zpráva znovu. Časový limit bude nastavený na $1500ms$ i s ohledem na to, že se MCU na měřicím zařízení bude moci uspat až na $1000ms$.

Pro dosažení velké efektivity přenosu bude zvolen bitově orientovaný přenos dat. Má výhodu, že se přenáší minimum přebytečných informací oproti znakově orientovanému. Tato vlastnost je důležitá při využití technologie BLE, prozože nemá velkou propustnost dat (viz sekce 2.3.5). Dalším faktorem je, že čím méně bude přenášeno dat, tím bude větší úspora energie.

Při návrhu protokolu byl uvažován i znakově orientovaný přenos. Při využití tohoto typu přenosu by bylo možné použít některý ze standardů pro kódování dat jako je například JSON, YAML, XML a další. S tímto přístupem by byl protokol jednoduchý na implementaci vzhledem k dostupnosti parsovacích knihoven ve vysokoúrovňových jazycích. Nevýhodou by byla ale vyšší režie při přenosu dat, protože každý z těchto standardů využívá řídící znaky. Řídící znaky pak mohou zabrat větší část datové zprávy než samotná data.

Návrh protokolu je popsán pomocí normy ASN.1 [5]. Navržené struktury zpráv jsou obsažené v příloze B. V návrhu jsou použity některé datové typy specifické pro tuto práci. V následujícím výčtu budou popsány datové typy, které jsou využity v návrhu a nejsou specifikovány normou ASN.1:

- **Byte** - 1 bajtové neznaménkové celočíselné číslo
- **Short** - 2 bajtové neznaménkové celočíselné číslo
- **Integer** - 4 bajtové neznaménkové celočíselné číslo

Zpráva se kóduje pravidly BER. To znamená, že každá zpráva má strukturu sestávající se z identifikátoru zprávy, délky zprávy a obsahu.

3.2.1 Hlavička protokolu

Všechny zprávy odesílané protokolem začínají základními informacemi. První z těchto informací je `operační kód` zabírající 1 bajt. `Operační kód` určuje typ posílané zprávy. `Délka zprávy` zabírající 2 bajty je další parametr, ta vyjadřuje počet bajtů za hlavičkou zprávy. Ve vývojové fázi bude zahrnuta na konec zprávy ukončovací značka `0x0A0D`. Ukončovací značka bude sloužit pro kontrolu, že probíhá správně serializace přijaté zprávy. Posledním parametrem v hlavičce je `časová značka`, pro kterou jsou vyhrazeny 4 bajty. Tato položka by měla obsahovat časovou hodnotu od zahájení komunikace mezi zařízeními v milisekundách. Očekává se, že klient pošle jako první hodnotu 0 a následně se hodnota zvyšuje o časový přírůstek. Pro hodnoty byly vyhrazeny pouze 4 bajty, protože místo umožňuje počítat časové značky necelých 50 dnů. Není předpokládáno, že zařízení bude spuštěno déle než pár dní i vzhledem k připojení na baterii, a proto toto místo stačí.

3.2.2 Zahajovací zpráva

Tata zpráva bude zahajovat zpracování dat na měřicím zařízení. To znamená, že nebude probíhat žádné zpracování dat moduly do té doby, než bude doručena tato zpráva na měřicí zařízení. Ty moduly, které nemusí být spuštěné, budou uspané. Pokud by sběrné zařízení poslalo zprávu s požadavkem o data (viz sekce 3.2.3) na měřicí zařízení bez předchozí zahajovací zprávy, bude mu odpovězeno zprávou záporného potvrzení (viz sekce 3.2.10). Přesto bude možné poslat na měřicí zařízení zprávu pro vyžádání informací o připojených modulech (viz sekce 3.2.3) a dostat odpověď. Po odeslání této zprávy se budou budit moduly, u kterých je to třeba a začnou se zpracovávat data. Na zprávu s požadavkem o data (viz sekce 3.2.3) bude po tomto kroku odpovězeno zprávou s daty (viz sekce 3.2.6).

3.2.3 Zpráva s požadavkem

Aby bylo možné z měřicího zařízení dostat data, bude potřeba na toto zařízení poslat zprávu s požadavkem o tato data. Tuto funkci budou zastávat zprávy s touto hlavičkou. Pro funkci budou nadefinované dva typy zpráv, které se rozlišují podle parametru *ReqType*.

Zpráva s požadavkem o informace

Jednoduchá zpráva sloužící jako podnět pro získání informací o připojených modulech ke sběrnému zařízení. Jako odpověď bude poslána zpráva s informacemi o modulech (viz sekce 3.2.4) z měřicího zařízení .

Zpráva s požadavkem o data

Zpráva pro získání naměřených hodnot z modulů. Zpráva bude obsahovat parametr *bufferData*. Když bude tento parametr nastaven na hodnotu 0, znamená to, že se budou posílat pouze poslední naměřená data datovou zprávou (viz sekce 3.2.6). V případě, že by byl parametr nastaven na jinou hodnotu než 0, je to podnět k tomu, že se mají poslat hodnoty celého bufferu naměřených hodnot z modulu, a to datovou zprávou odesílající všechna data z bufferu (viz sekce 3.2.7). Z jakých modulů budou data získávat je určeno výčtem *identifikátorů* jednotlivých modulů.

3.2.4 Informativní zpráva

Tato zpráva slouží pro informování klientského zařízení, jaké moduly jsou dostupné na sběrném zařízení a jak jsou nastaveny nebo na jaké hodnoty je lze nastavit. Jako první parametr je ve zprávě obsažen parametr *ID modulu*. Tato informace je velice důležitá, protože *ID modulů* se používá ve zprávě pro vyžádání dat (viz sekce 3.2.3). V případě tvorby vlastní aplikace implementující tento protokol může tato informace sloužit pro získání *ID dostupných modulů*. Kdyby nebyla tato zpráva na toto použita, musí být *ID modulů* zjištěno ze specifikace.

Dalším parametrem je *typ modulu*. *Typ modulu* je určen číselnou hodnotou,

kdy každou unikatní hodnotou je definováno, jaký modul představuje. Jsou definovány následující moduly:

1. modul akcelerometru
2. modul gyroskopu
3. modul teploměru
4. modul měření tepu
5. GPS modul
6. pohybový modul

Vzhledem k vyhrazení jednoho bytu pro určení tohoto typu je možnost v budoucnu dodefinovat další typy modulů, nebo naopak nepotřebné ubrat.

Další tři parametry informují o tom, v jakém časovém intervalu se na sběrném zařízení spouští proces na zpracování dat. To, jak často se tento proces spustí odpovídá době uspání hlavní smyčky krát počet vynechaných intervalů nastavených na modulu.

Z parametru *sleep* se dá vyčíst, jestli je modul uspaný nebo ne. Uspaným modulem je myšleno, že se u modulu nezpracovávají hodnoty do bufferu a pokud je to možné, je modul uspán. Je potřeba dávat pozor jaký modul se má uspat nebo jaký už spí, protože některé moduly jsou na sobě závislé. Například pokud se uspí modul akcelerometru, uspí se s ním zároveň i modul měřící intenzitu pohybu. Modul k měření intenzity pohybu je závislý na akcelerometru (viz sekce 3.1.2). Posledními dvěma parametry jsou popis a délka popisu. Z těchto hodnot se dá vyčíst krátký informativní řetězec o modulu. Délka řetězce je omezena na 16 znaků.

3.2.5 Zpráva nastavení

Měřicí zařízení je možné nastavovat. Nastavení měřicího zařízení bude realizováno tímto typem zpráv.

Zpráva nastavení modulů

Zpráva se posílá ze sběrného zařízení na měřicí zařízení. Jejím účelem bude nastavit moduly připojené k měřicímu zařízení. Aby se mohly pomocí této zprávy nastavit moduly, je třeba znát hodnoty, na které je možné nastavovat parametry modulu. Dále je potřeba znát *ID modulů*, které jsou k zařízení připojeny. Toho se dá docílit pomocí vyžádání zprávy s informacemi (viz sekce 3.2.3) nebo ze specifikace měřicího zařízení.

Zpráva je podtyp zpráv pro nastavení měřicího zařízení a v hlavičce tohoto podtypu zpráv je označena hodnotou 1. Samotná zpráva se pak sestává z výčtu 1 až 255 modulů, kdy obsahuje tři parametry. *ID modulu*, kterého se týká nastavení. *Interval čtení* sloužící pro nastavení intenzity zpracování dat z modulů (viz sekce 3.2.4). Poslední hodnotou je *sleep*, který modul na sběrném zařízení vzbudí, uspí nebo neudělá nic, pokud se pošle hodnota, na kterou je již nastaven. V případě nastavení hodnoty *sleep* na 0 se modul na měřicím zařízení vzbudí. V opačném případě, kdy je zadána hodnota z rozsahu 1-255, je modul uspán.

Ověřením, že proběhlo nastavení modulů pomocí této zprávy úspěšně, je přijetí zprávy kladného potvezení (viz sekce 3.2.9). V opačném případě, když bude jeden ze tří parametrů obsahovat nevalidní hodnotu, odešle se zpráva záporného potvrzení (viz sekce 3.2.10).

Zpráva nastavení hlavní smyčky

Zpráva je podtypem zpráv pro nastavení sběrného zařízení a hodnotou v hlavičce pro nastavení je 2.

Sběrné zařízení má kvůli úspoře energie nastavené uspání procesoru na určitou časovou hodnotu. Ve výchozím nastavení je tato hodnota nastavena na 100 milisekund. Zasláním této zprávy dojde k přenastavení této hodnoty na hodnotu z intervalu 50-1000 milisekund.

V případě, že na zařízení došla zpráva korektně a hodnota je z intervalu, je odeslána zpět zpráva typu ACK v opačném případě NACK.

3.2.6 Datová zpráva

Pomocí této zprávy se posílají data získaná z měřicího zařízení do sběrného zařízení. Data z modulů jsou členěna jako celek odpovídající nějakému modulu. Každá část dat získaná z některého modulu je uvozena *ID modulu*. Další parametr říká, kolik dat se z modulu přeneslo. Od této části zprávy se začínají samotná data z modulu. První část této části dat říká, o jaký datový typ se jedná. V protokolu jsou nadefinovány všechny základní celočíselné, desetinné a logické datové typy, kdy každý z těchto typů má přiřazenou celočíselnou hodnotu z rozsahu 1-255. Když známe typ dat, můžeme přečíst blok dat, který se může skládat z 1 až 8 bajtů. Délka datového bloku se odvodí podle typu dat. Poslední parametry v datovém bloku popisují veličinu, ke které se data vztahují.

Zpráva se posílá ze sběrného zařízení jako odpověď na zprávu s požadavkem o data (viz sekce 3.2.3). Tato zpráva se sestaví na základě přijatých *ID modulů*. V odpovědi jsou pak moduly seskládány za sebou tak, jak přišly ve zprávě se žádostí o data.

Důvod navržení přenášení dat tímto způsobem má za účel co nejvíce zmenšit počet informací, které je potřeba přenést. To se zajišťuje volným počtem bajtů tvořících datovou část. Například u akcelerometru jsou přenášeny tři hodnoty popisující gravitační zrychlení. Oproti teploměru, kde je přenášena pouze jedna hodnota.

3.2.7 Datová zpráva se všemi daty z bufferu

Tato zpráva vychází z datové zprávy (viz sekce 3.2.6). Rozdíl mezi zprávami je, že v datové zprávě je poslána poslední získaná hodnota z modulu. Oproti tomu v této zprávě je posláno až 5 posledních získaných hodnot. Hodnoty jsou seřazeny pro každý modul od nejnovější po nejstarší. Zpráva se využije například, když dojde k přerušení spojení mezi měřicím zařízením a sběrným zařízením. Přijetím této zprávy se získá alespoň část dat, která nemohla být získána z důvodu přerušení spojení.

Zpráva se odesílá z měřicího zařízení na základě přijetí zprávy s požadavkem na data (viz sekce 3.2.3), kde parametr *bufferData* této zprávy je nastaven na hodnotu z intervalu 1 až 255.

3.2.8 Ukončovací zpráva

Touto zprávou se zastaví zpracovávání dat na měřicím zařízení. Po přijetí se zastaví proces získávání dat a uspí se moduly. Uspávají se ty moduly, které to umožňují.

3.2.9 Zpráva kladného potvrzení

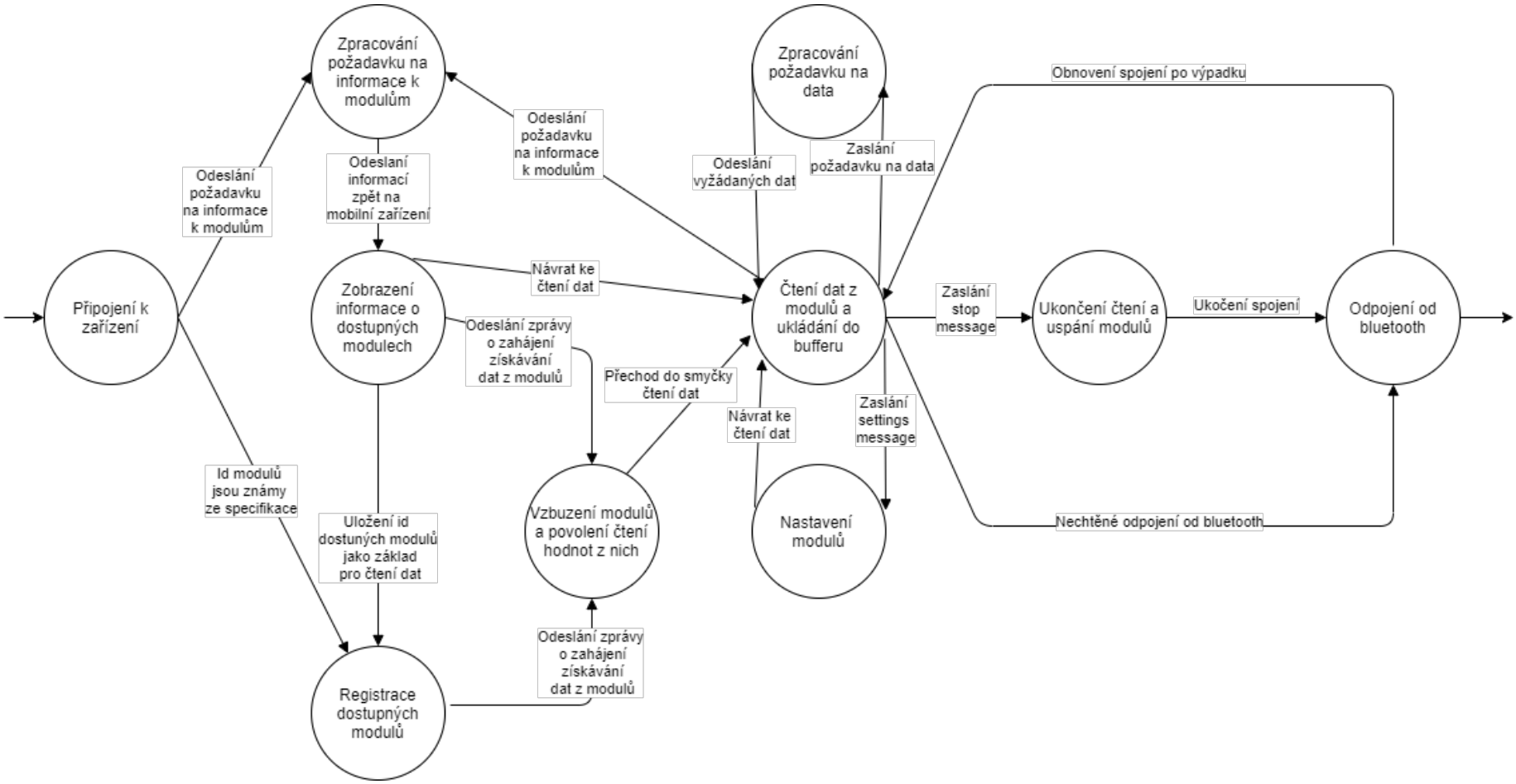
Zpráva tohoto typu se posílá vždy jako kladná odpověď, když byla přijatá zpráva zpracována bez problému. Odesílána je vždy jen z měřicího zařízení na připojené sběrné zařízení. Kromě zpráv, kdy jsou vyžadována data nebo informace, se odesílá jako odpověď na ostatní typy zpráv.

3.2.10 Zpráva záporného potvrzení

Opak zprávy kladného potvrzení. Odesílá se, když nebylo možné přijatou zprávu zpracovat. Může indikovat chybu v přenášených datech. Posílá se i při nevalidní hodnotě, kterou zařízení přijme a není možné jí zpracovat.

3.2.11 Stavový diagram připojení

Na obrázku 3.2 lze vidět stavový diagram připojení. Jde vlastně o konečný automat popisující, jak vypadá průběh připojení sběrného zařízení k měřicímu zařízení pomocí protokolu.



Obrázek 3.2: Stavový diagram průběhu komunikace měřicího a sběrného zařízení

3.3 Aplikace sběrného zařízení

Navrhovaná aplikace se bude umět připojit k měřicímu zařízení a komunikovat s ním pomocí navrženého protokolu (viz sekce 3.2). Kromě toho bude v této aplikaci jednoduchý výpis přijatých dat. Do aplikace bude implementováno nastavení měřicího zařízení pomocí protokolu (viz sekce 3.2).

Aplikace bude vytvořena pro platformu Android.

3.3.1 Připojení k měřicímu zařízení

Modul na měřicím zařízení podporuje BLE, ale i Bluetooth s vyšší spotřebou energie. Mobilní aplikace navržená v této práci bude podporovat pouze nízkoenergetickou verzi BLE. V případě, že by se v budoucnu navrhovala jiná aplikace pro jakýkoliv typ zařízení, je možné pro přenos dat využít i Bluetooth.

Bude potřeba naimplementovat několik funkcí pro komunikaci s BLE. Jednou z funkcí bude skenování dostupných BLE zařízení v okolí mobilního zařízení. Každé nalezené zařízení podporující BLE se zobrazí v aplikaci. U každého z nalezených zařízení bude zobrazen název, MAC adresa a tlačítko pro připojení. Proces připojení bude obsahovat spárování zařízení, zapnutí notifikace na charakteristice BLE modulu a odeslání zahajovací zprávy (viz sekce 3.2.2) pro zahájení zpracování dat. Díky povolení notifikací na BLE charakteristice nebude nutné v mobilní aplikaci zkoušet číst data z charakteristiky bez zaručení, že už tam jsou data připravena. Při aktivaci notifikací jsou data rovnou posílána na mobilní zařízení. Tímto přístupem se i zkrátí doba přenosu dat z měřicího zařízení na připojené sběrné zařízení. Poslední fází připojování bude odeslání zahajovací zprávy (viz sekce 3.2.2). Jakmile na ni přijde odpověď, odemknou se mechanismy pro získávání dat z měřicího zařízení.

Jak bylo popsáno výše, čtení přijatých dat bude probíhat na základě notifikace od BLE modulu sběrného zařízení. Oproti tomu zápis probíhat jednoduchým zápisem data na dostupnou charakteristiku.

3.3.2 Modul pro SmartCGMS

Aplikace bude sbírat data z měřicího zařízení a ukládat si je do své vnitřní paměti. Aplikace bude komunikovat s modulem v systému SmartCGMS prostřednictvím síťového protokolu, pomocí kterého bude přenášet do systému SmartCGMS uložená data.

3.3.3 Ukládání přijatých dat

Není zaručeno, že data přijatá ze sběrného zařízení budou moci být okamžitě poslána do SmartCGMS. To může být způsobeno tím, že uživatel nebude mít možnost se připojit k serveru předávajícímu data SmartCGMS. Pro tyto potřeby se budou data ukládat do souborů na mobilním zařízení. Každý modul, ze kterého se budou dostávat data, bude mít vlastní soubor pro data. Data budou ukládána ve formátu CSV. Formát bude mít strukturu *název veličiny*, *hodnota veličiny*, *čas získání hodnoty*.

Tyto soubory budou sloužit pro nahrání těch dat, která nebude možné do SmartCGMS dopravit přes síť.

3.4 Filtr pro ukládání dat do SmartCGMS

V úvodu bylo zmíněno, že architektura SmartCGMS se sestává z filtrů propojených za sebou. Pro ukládání vytěžených dat sběrným zařízením z měřicího zařízení bude vytvořen nový filtr do SmartCGMS. Tento filtr bude síťovým serverem pro přijetí dat odeslaných z mobilní aplikace pomocí nadefinovaného protokolu (viz sekce 3.5). Přijatá data se pomocí filtru zpracují do nové události a pošlou se do následujícího filtru.

3.5 Protokol pro přenos dat do SmartCGMS

Architektura SmartCGMS nebude zahrnuta v mobilní aplikaci. SmartCGMS bude fungovat jako síťová služba dostupná ze sítě Internet. Bude navržen protokol nad TCP/IP, který zajistí výměnu dat mezi aplikacemi. Přenos dat bude realizován nad trasportním protokolem TCP.

Návrh protokolu je popsán pomocé normy ASN.1 [5]. Navržené struktury zpráv jsou obsažené v příloze C.

3.5.1 Hlavička protokolu

Hlavička se bude skládat ze tří základních částí. První část bude tvořit 4 bajtový indentifikátor hlavičky. Ten bude identifikovat, že se jedná o zprávu protokolu. Tato část zprávy poslouží i k vyfiltrování nechtěných zpráv posílaných na server. Další částí bude typ zprávy. Typ zprávy si z hlavičky zabere 1 bajt dat. Protokol bude umět posílat pouze jeden typ zprávy, a to datovou zprávu (viz sekce 3.5.2). Tento typ bude identifikován číselnou hodnotou 1. V protokolu se nebudou používat žádné ukončovací znaky. I z tohoto důvodu bude posledním parametrem hlavičky délka těla zprávy. Hodnota *délky těla* zprávy představuje délku zprávy od konce hlavičky zprávy v bajtech. Pro *délku těla* zprávy jsou vyhrazeny 2 bajty. To znamená, že maximální délka zprávy je omezena na 65535 bajtů.

3.5.2 Datová zpráva

Základem datové zprávy je struktura jednoho datového bloku. Tento datový blok obsahuje identifikátor typu přenášených dat. V současné chvíli je nedefinováno šest typů dat, která se mohou přenášet:

- `Temperature_Skin` (1) - Naměřená teplota lidského těla
- `Temperature_Ambient` (2) - Naměřená teplota okolí
- `Position_Lat` (3) - Zeměpisná šířka
- `Position_Lon` (4) - Zeměpisná délka
- `Movement` (5) - Naměřená intenzita pohybu

Jak je vidět z výčtu, každý z datových typů má svou přiřazenou hodnotu, která slouží jako identifikátor. Pro identifikátor je vyhrazen jeden bajt. Další částí v datovém bloku je časová značka. Časová značka má formát Unixové časové značky. Hodnota časové značky tedy zobrazuje počet sekund od 1.1.1970. Pro časovou značku jsou využity čtyři bajty místa. Poslední

hodnotou je samotná přenášená hodnota. Pro hodnotu je vyhrazeno osm bajtů. U této hodnoty se předpokládá, že bude v datovém formátu double. Všechny číselné hodnoty, které se budou přenášet, i když budou mít celočíselný tvar, by se měli převést nejdříve do tohoto formátu. Datovou část ve zprávě může tvořit až 5041 datových bloků reprezentujících naměřené veličiny. Tato hodnota plyne z maximální velikosti specifikované délky zprávy v hlavičce protokolu (viz sekce 3.5.1).

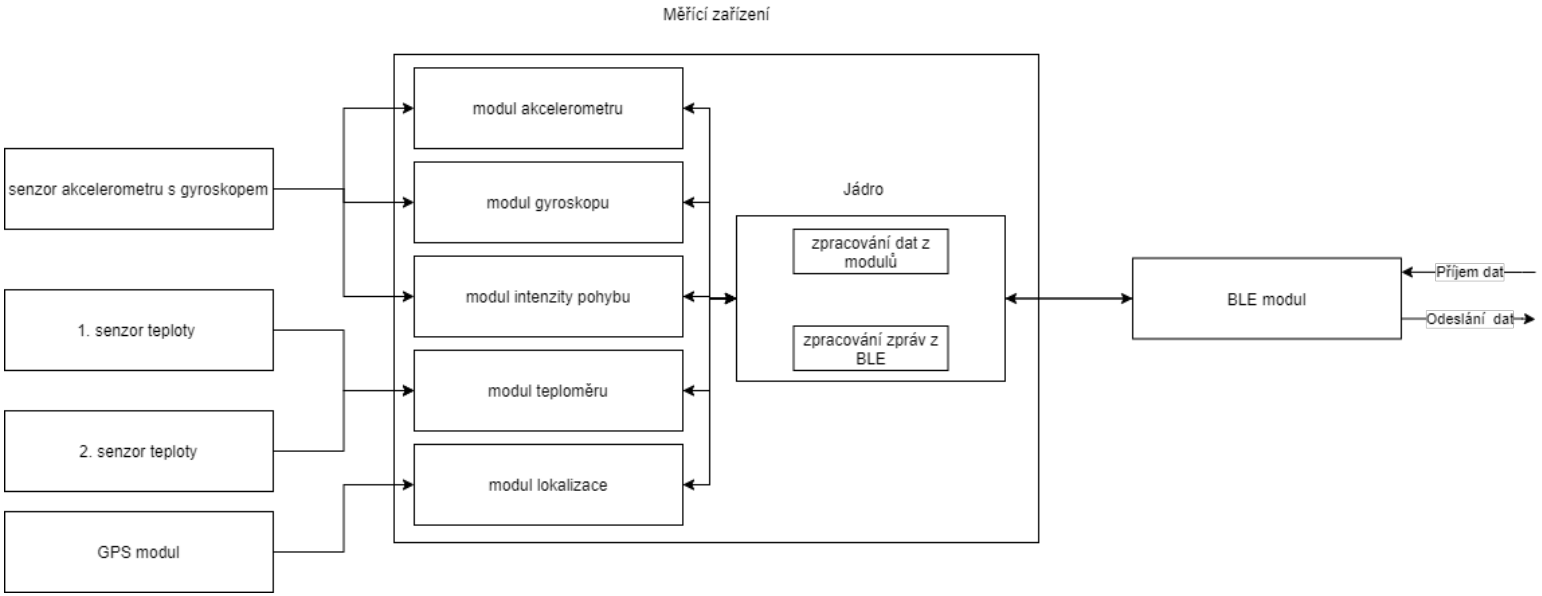
4 Implementace

V následující kapitole bude popsána implemtace měřicího zařízení, aplikace sběrného zařízení a filtru pro SmartCGMS.

4.1 Firmware pro měřicí zařízení

V následující sekci je popsána implementace firmware měřicího zařízení. Firmware se skládá z implementace jádra, modulů a protokolu pro přenos dat pomocí technologie BLE. Struktura firmware, a běh dat mezi jednotlivými částmi, je znázorněna na obrázku 4.1.

Na použité desce STM32F103C8T6 není optimálně vyřešena práce s desetinnými čísly. Desetinná čísla se na desce dopočítávají softwarově a pouze s omezenou přesností, kdy lze desetinné číslo reprezentovat pouze 4 bajty. Na desce je tedy možné pracovat s desetinnými čísly, ale kvůli jejich softwarovému zpracování, je to časově náročné a nepříliš přesné. Z tohoto důvodu je v implementaci co nejvíc omezena práce s desetinnými čísly. U modulů vracejících hodnotu v podobě desetinného čísla, se tato hodnota převede na celé číslo s pevnou desetinnou čárkou. To se provede vynásobením hodnoty číslem 10 umocněným na požadovaný počet desetinných míst.



Obrázek 4.1: Struktura firmware měřicího zařízení

4.1.1 Jádro

Hlavní řídící kód firmwaru se nachází v souboru `modules.c`. Je zde několik funkcí, které zprostředkovávají komunikaci se senzory, které se dále dělí na moduly:

`modules_main_loop`: Hlavní funkce kódu, která jako jediná je spouštěna z hlavní smyčky programu. Tato funkce se stará o volání funkcí modulů pro zpracování dat. Pokud přijde zpráva na BLE, tato funkce má v sobě podmínku s podmínkovou proměnou `modules_message_readed_flag`, která se nastaví, jakmile je dokončeno čtení zprávy. V bloku této podmínky začne zpracování přijaté zprávy.

`modules_init_all_modules`: Hlavním účelem této funkce je zavolat inicializační funkci všech modulů. Funkce se volá pouze při startu zařízení.

4.1.2 Protokol

K zajištění komunikace pomocí technologie BLE je k zařízení připojen modul JDY-33 (viz sekce 3.1.5). Komunikace s tímto modulem je realizována pomocí rozhraní UART, na měřicím zařízení konkrétně jde o rozhraní UART1. JDY-33 má pouze jednu charakteristiku pro příjem a odeslání dat. Přijímání dat se provádí pomocí přerušení na rozhraní UART1. Pokud jsou nějaká data pro čtení k dispozici, vyvolá se přerušení a po jednotlivých znacích se začnou zpracovávat data. Data se ukládají do dočasného bufferu `message_buffer`, který je definován v souboru `modules.c`. Při zpracování se pracuje s několika příznaky a dočasnými proměnnými:

- `modules_message_reading_flag` – Pokud je proměnná nastavena na hodnotu 1, právě probíhá čtení. Když je nastavena na jinou hodnotu, žádná data se nečtou. Proměnná slouží pro kontrolní mechanismus, zda dorazila zpráva v pořádku. Jakmile přijde první znak zprávy, spustí se časový odpočet. Není-li zpráva přijata do časového limitu (např. ztráta spojení), mechanismus vyresetuje všechny důležité proměnné spojené s přijímáním zpráv.
- `modules_message_readed_flag` – Proměnná určující, jestli byla přečtena všechna data z charakteristiky. Při úspěšném čtení dat se nastaví

hodnota na 1. Je-li nastavena proměnná na hodnotu 1, ve funkci `modules_main_loop` se spustí mechanismus zpracování zprávy uložené v `message_buffer`. Po jeho ukončení je proměnná vyresetována na hodnotu 0.

- `modules_message_len_par` – Při čtení hlavičky zprávy v mechanismu přerušení se nastaví do tohoto parametru přijatá délka těla zprávy. Ta slouží pro přijetí odpovídajícího počtu bajtů následujícím za hlavičkou zprávy.
- `modules_message_buffer_index` – Proměnná udávající aktuální index zpracovávaného znaku v proměnné `message_buffer`.

Když je přijata celá zpráva, její zpracování probíhá pomocí funkcí v souboru `protocol.c`. Zde se v první řadě zkontroluje hlavička protokolu, zda obsahuje korektní hodnoty *operačního kódu*, *délky zprávy* a *časové značky*. Proběhne-li kontrola dobře, hodnoty se uloží pro další zpracování. Kdyby byla nalezena chyba v hlavičce, ukončí se další zpracovávání zprávy a pošle se zpráva záporného potvrzení (viz sekce 3.2.10). Po úspěšné kontrole se zjišťuje, jakého typu je přijímaná zpráva. Typ se určuje podle *operačního kódu* a případně *operačního kódu podtypu zprávy*.

Zahajovací zpráva

Po spuštění měřicího zařízení nejsou zpracovávána data z modulů, protože je to zakázáno ve funkci `modules_main_loop`, kde se toto volá v podmínkovém bloku závislém na logické proměnné `modules_message_data_procesing_flag`. Přijetí této zprávy neudělá nic jiného, než že nastaví logickou proměnou na hodnotu 1, čímž se spustí získávání dat z modulů.

Ukončující zpráva

Je opakem zahajující zprávy. Při přijetí nastavuje logickou proměnnou `modules_message_data_procesing_flag` na hodnotu 0 a tím zastavuje zpracování dat z modulů.

Zpráva nastavení intervalu hlavní smyčky

Z přijaté zprávy se vezme časová hodnota, na jakou má být nastaveno uspávání hlavní smyčky. Pokud je hodnota ze stanoveného rozsahu $50ms$ až $1000ms$ a je přenastavena proměnná `modules_main_loop_interval` nesoucí v sobě časovou hodnotu uspání hlavní smyčky. V případě, že je přijatá hodnota mimo stanovený rozsah, nic se nenastavuje a pomocí zavolání funkce `protocol_send_ACK_message`, se nastaví parametr `message_type` na hodnotu 8. Poté se odešle zpráva záporného potvrzení.

Zpráva nastavení modulů

Tato zpráva obsahuje data s nastavením *uspání* a *intervalu čtení* modulů. Ze zprávy se vytáhnou informace pro jednotlivé moduly a tyto informace se předají funkci `modules_set_modules`, kde se nastaví moduly na přijaté hodnoty. Kdyby byla některá z hodnot mimo rozsah, vrací funkce hodnotu 0 a to vyvolá odeslání zprávy záporného potvrzení. Nastaví-li se všechny moduly v pořádku, pošle se zpráva s kladným hodnocením. Odeslání zprávy záporného nebo kladného potvrzení zprostředkuje funkce `protocol_procces_module_setting_message`, kde se vyhodnotí návratová hodnota z funkce `modules_set_modules`.

Zpráva s požadavkem na data

Zpráva s požadavkem na data obsahuje ve svém těle ID modulů, ze kterých se mají odeslat data. Ve funkci `protocol_process_module_request_data_message` se získaná ID předají funkci `modules_get_data`. Funkce `modules_get_data` k ID jednotlivých modulů získá příslušné ukazatele na struktury `Module_t`. Ve funkci se pomocí ukazatelů získá přístup k funkci `read`, která se zavolá pomocí všech získaných ukazatelů. Funkce `read`, vrací přes pointer v parametru funkce poslední získanou hodnotu. Ze získaných dat se sestaví datová zpráva (viz sekce 3.2.6) ve funkci `modules_get_data`, která se po návratu z funkce odešle ve funkci `protocol_process_module_request_data_message`.

Zpráva s požadavkem na všechna data

Zpracování zprávy nebylo implementováno, ale v kódu jsou připraveny mechanismy pro rozšíření firmware a implementaci této zprávy.

Zpráva s požadavkem na informace

Přijatou zprávu zpracovává fuknce `protocol_process_module_request_info_message`. Ve funkci se volá funkce `modules_get_module_info`, která získá informace o všech definovaných modulech ve firmware a vrací přes pointer v parametru tělo zprávy (viz sekce 3.2.4), kde jsou zahrnuty získané informace. Po té, co je tělo zprávy vráceno do fuknce `protocol_process_module_request_info_message`, dosestaví se informativní zpráva (viz sekce 3.2.4) a je odeslána. Získávají se informace o ID, *typu zařízení*, *výchozím intervalu čtení*,*minimálním intervalu čtení*, *maximálním intervalu čtení*, *uspání*, *délce popisu* a *popisu*.

4.1.3 Moduly

Modulem je v implementaci myšlena abstraktní datová struktura, která udržuje data jedné veličiny. Například data z akcelerometru, jak je znázorněno na obrázku 4.1. Implementace každého modulu by měla zahrnovat základní množinu funkcí definovaných ve struktuře `Module_t` viz níže. Pokud to podoba dat získaných z modulu umožňuje, měl by modul mít kruhový buffer pro uložení hodnot.

Implementace kruhového bufferu a funkcí ze struktury `Module_t` se může lišit pro každý modul, a proto jsou tyto implementační detaily popsány pro každý modul v podsekcích této sekce.

Pro moduly byla vytvořena datová struktura `Module_t`. Datová struktura je navržena, aby pokryla všechny požadavky na operaci s moduly. Je vytvořena i s přihlédnutím dalšího možného vývoje popisovaného firmware. Struktura se sestává ze dvou částí.

V první části jsou zahrnuty parametry. Parametry `name` a `dev_type` plní pouze informativní roli při zasílání zprávy (viz sekce 3.2.4), kdy se zpráva plní i hodnotami z těchto parametrů. Parametr `module_id` je klíčový pro ce-

lou řadu operací s modulem. Slouží jako jednoznačný identifikátor a je třeba, aby byl znám při posílání požadavků na data z modulu. Tato informace se dá získat posláním zprávy s požadavkem o informace (viz sekce 3.2.3) a následně zpracováním informativní zprávy (viz sekce 3.2.4). Případně se ID modulů mohou také získat z podsekcí s popisem implementace jednotlivých modulů. Parametry `current_read_interval`, `min_read_interval`, `max_read_interval` slouží pro nastavení intervalu zpracování dat modulu. Jak napovídají názvy, první ze zmíněných obsahuje nastavenou hodnotu a zbylé dvě hodnoty na jakou nejmenší a největší hodnotu může být parametr `current_read_interval` nastaven. Tato hodnota určuje za jaký časový interval proběhne zavolání funkce `process_data`. Doba mezi voláními fukce pro zpracování se vypočítá vynásobením hodnoty `modules_main_loop_interval` s hodnotou `current_read_interval`. Posledním parametrem je `sleep`. Parametr sleep slouží jako logická proměnná. Jeho hodnota rozhoduje o tom, jestli se bude nebo nebude volat funkce `process_data`.

V druhé části jsou funkce. Funkce `init` se stará o základní nastavení modulu. Pokud se senzor dělí na více modulů, je možné, že bude potřeba mít pro moduly společné části kódu ve funkci `init`. Jednou možností je vytvoření společné části `init` pro každý modul. Společné se ošetří podmínkami, aby nebyly volány vícekrát. Druhou možností je implementovat `init` pouze pro jeden modul a ostatním nechat funkci `init` na hodnotu `null`. Funkce `read` slouží k přečtení poslední hodnoty uložené v kruhového bufferu. Funkce `read_all` čte všechny hodnoty z kruhového bufferu. Tato funkce nebyla kvůli časové limitaci implementována. V případě zápisu dat na modul je tu funkce `write`. Funkce `write` nebyla využita v implementaci u žádného z modulů. Funkce ale byla definována pro případné budoucí rozšíření měřicího zařízení o další modul, kde může být potřeba i zapisovat data. Poslední dvě definované funkce jsou `process_data` a funkce `sleep_func`. Funkce `process_data` zpracuje data z modulu a ukládá hodnotu do kruhového bufferu. Funkce `sleep_func` uspává modul pokud to jeho implementace funkce umožňuje.

Akcelerometr

Implementace modulu akcelerometru se nachází v souboru `mpu9250.c`. Modul získává data ze senzoru MPU9250. Pro modul je vytvořený kruhový buffer `circular_buffer_accel`, kam se ukládají naměřené hodnoty. Kruhový buffer má vyhrazené místo pro 5 hodnot. Modul akcelerometru implementuje ze struktury `Module_t` funkce, `init`, `read`, `process_data` a `sleep_func`.

Komunikace modulu se senzorem probíhá pomocí rozhraní I2C. Skrze toto rozhraní probíhá nastavení senzoru a zejména části, která se týká akcelerometru. Nastavuje se rozsah zrychlení, které senzor bude snímat a celkové nastavení senzoru MPU9250. Rozsah zrychlení se nastavuje úpravou registru `0x1C`. V implementaci je nastaven rozsah na `2G`. Nastavení se provede posláním hodnoty 0 na zmíněný registr senzoru přes I2C rozhraní.

Zpracování dat začíná přečtením šesti 8bitových registrů senzoru MPU9250 týkajícího se akcelerometru. Data začínají v registru s adresou `0x3B`. Čtení dat probíhá zavoláním knihovní funkce `HAL_I2C_Mem_Read` která přečte od počáteční adresy 6 bajtů dat z registrů senzoru. Přijatá data se postupně ukládají do 2 bajtových celočíselných proměnných (`int16_t`). První bajt, co přijde, se uloží do levé části 2 bajtové proměnné a druhý do pravé části. Toto se opakuje pro zbylé dvě proměnné. Hodnoty všech tří přijatých proměnných jsou nezpracovaná data.

Data je potřeba převést na jednotky gravitačního přetížení, kvůli správné fukci modulu pro měření intenzity pohybu. Převod proběhne vynásobením přijaté hodnoty škálovacím faktorem. Škálovací faktor je odvozen od nastavení přesnosti měření. Hodnota škálovacího faktoru se dá zjistit z manuálu výrobce nebo výpočtem $scalingFactor = numericRange/scaleRange$. $numericRange$ je číselný rozsah který dokáže měřit senzor a ten je 65535 hodnot. $scaleRange$ je nastavený rozsah měření v jednotkách gravitačního přetížení $G * 10^{-4}$. Nyní pomocí škalovacího faktoru vynásobíme každou naměřenou hodnotu škálovacím faktorem a získáme jednotky gravitačního přetížení v jednotkách G. Získaná data mají desetinnou podobu a kvůli důvodům zmíněným v sekci 4.1 je převedeme na celá čísla s pevnou desetinnou čárkou o třech desetinných místech. Když je celá tato operace provedena, uloží se data do kruhového bufferu.

Na popsaném procesu je závislý pohybový senzor. Na konci procesu zpracování dat v modulu akcelerometru se vezmou získané hodnoty a zpracovává se z nich intenzita pohybu. Hodnoty jsou vlastně vektory. Z vektorů se vypočte celkový vektor zrychlení. Z celkových vektorů zrychlení (aktuálního a přechozího získaného) se udělá rozdíl a hodnoty rozdílu se započítají do průměru. Po deseti změřených hodnotách se celkový průměr vyhodnotí a podle tabulky 3.1 se určí intenzita pohybu. Získaná hodnota se ukládá do bufferu pro pohybový senzor.

Uspání modulu probíhá zápisem do 1 bytového registru `0x6C`. Pátý až třetí bit (5 = x, 4 = y, 3 = z) v registru nastavuje, jestli se měří data na ose.

Když má bit hodnotu 1, data se na ose neměří, naopak při 0 probíhá měření dat.

Gyroskop

Funkčnost je velmi podobná modulu akcelerometru. Implementace se nachází v souboru `mpu9250.c`. Rozdíly jsou jen malé a týkají se rozdílné povahy získaných dat. Modul implementuje stejnou čtveřici metod jako akcelerometr. Pro gyroskop je vytvořen kruhový buffer `circular_buffer_gyro` s vyhrazeným místem na 5 hodnot.

I pro tento modul se používá komunikace I2C se senzorem MPU9250. Než se začnou získávat a zpracovávat data, probíhá nastavení MPU9250. Nastavuje se část operující s gyroskopem. Nastavuje se rozsah měření úhlové rychlosti. Rozsah je nastaven na hodnotu 250 stupňů za sekundu. Hodnota tohoto rozsahu se nastavuje zápisem pomocí I2C hodnoty 0 na registr `0x1B` senzoru MPU9250.

Data se získávají a zpracovávají podobně jako u akcelerometru ze šesti 8bitových registrů. Začíná se číst od registru s adresou `0x43`. Data získaná z registru se zpracovají stejným postupem, jak bylo popsáno u akcelerometru. Rozdílná je hodnota škálovacího faktoru. Ta se získá výpočtem $scalingFactor = \frac{1}{numericRange/scaleRange}$. Hodnota $numericRange$ je 65535, protože se ze senzoru získávají dvoubitová celá čísla. Hodnota $scaleRange$ range je 500, protože senzor měří hodnoty na intervalu (−250,250). Výsledná hodnota po dokončení převodu je úhlové zrychlení ve stupních za sekundu.

Modul se uspává stejným procesem jako akcelerometr. Rozdíl je, že nastavuje druhý až nultý bit (2 = x, 1 = y, 0 = z).

Teploměry

Základem tohoto modulu jsou dva senzory TMP36. Data z teploměrů se získávají pomocí ADC převodníku. Pro získání hodnot je použit jeden ADC převodník, na kterém se využívají dva kanály. Jeden kanál se rovná jednomu senzoru TMP36. Pro co největší přesnost dat je nastavena vzorkovací hodnota na co nejvyšší úroveň na obou použitých kanálech. Hodnota je nastavena na 239,5 vzorků za sekundu. To je největší hodnota, kterou umožňuje nastavit mikrokontroler. Nejvyšší hodnota je nastavená z toho důvodu,

že při nižším počtu vzorků byl velký rozptyl vrácených hodnot v krátkých časových intervalech a kvůli tomu bylo měření méně přesné.

Kód modulů se nalézá v souboru `tmp36.c`. Modul ze struktury `Module_t` funkce, `init`, `read`, `process_data` a `sleep_func`.

Získání dat ze senzoru začíná tak, že se zavolá knihovní funkce `HAL_ADC_Start_DMA` na získání dat pomocí ADC, kdy data nejdou skrze MCU do paměti, ale pomocí řadiče DMA. Je nastaveno že se má získat z každého kanálu sto hodnot. Jakmile se získá požadovaný počet hodnot z obou kanálů, začne proces zpracování dat. Ze sta získaných hodnot se udělá aritmetický průměr. Hodnota aritmetického průměru je průměrná naměřená hodnota ADC. ADC pracuje na intervalu (0,4096), kdy maximální hodnota představuje $3.3V$. Hodnota napětí se získá vztahem $voltage = \frac{avargeADC}{4096}$. Hodnota napětí poslouží pro získání teploty ve stupních celsia. Teplota se získá dosazením do vztahu $temperature = (voltage - 0.5) * 100$. Výsledná hodnota se ukládá do kruhového bufferu jako 2 bytová celočíselná znaménková hodnota s pevnou desetinnou čárkou s jedním desetinným místem.

Funkce *sleep* definovaná není (viz sekce 3.1.3).

GPS

Základem pro modul je senzor NEO6M. Komunikace s modulem je realizována pomocí rozhraní UART. Na desce je pro to vyhrazený UART2. NEO6M používá protokol NMAE 0183 ve verzi 2.3 pro přenos dat pomocí UART. Struktura protokolu a jeho zpráv je k nalezení v manuálu výrobce [22].

Implementace modulu akcelerometru se nachází v souboru `neo6m.c`. Pro modul je vytvořený kruhový buffer `circular_buffer`, kam se ukládají naměřené hodnoty. Kruhový buffer má vyhrazené místo pro 5 hodnot. Modul GPS implementuje ze struktury `Module_t` funkce, `init`, `read`, `process_data` a `sleep_func`.

Pro získávání dat ze zařízení není potřeba posílat na senzor žádné zprávy. Data jsou posílána ze senzoru automaticky. Stačí zapnout čtení dat na příslušném UART rozhraní.

V inicializační metodě je provedena pouze inicializace kruhového bufferu na nulové hodnoty. Je to z toho důvodu, aby bylo snadnější poznat, že ještě

nebyla nalezena aktuální GPS souřadnice. Souřadnice se může zaměřovat i několik minut. V případě, že by byl senzor uvnitř budovy, nebo by byl jinak zastíněn signál ze satelitů, senzor nebude schopen určit souřadnice.

Proces získávání dat ze senzoru začíná tím, že se vyšle požadavek na DMA řadič pro přijetí dat z UART2 knihovní funkcí `HAL_UART_Receive_DMA`. Při tomto procesu se získají všechny zprávy zasílané senzorem. Ze všech zpráv se vyfiltruje zpráva s hlavičkou GLL. GLL je zpráva protokolu, ze které se dá získat zeměpisná šířka, délka a koordinovaný světový čas. U zprávy se kontroluje, jestli má odpovídající délku. Pokud ne pravděpodobně ještě nebyly získány souřadnice. Zpráva má pevnou délku, pokud obsahuje všechna očekávaná data.

Když je přijata zpráva s očekávanou délkou, začne proces zpracování dat. Data jsou přijata v podobě řetězce. Jako první se kontroluje kontrolní součet, zda nedošlo k chybě při přenosu dat. Kontrolní součet je vždy na konci zprávy a sestává se ze dvou hexadecimálních znaků v kodování ASCII. Proběhla-li kontrola úspěšně, ze zprávy se dostanou všechny tři hodnoty. Zeměpisná šířka je přijata ve formátu `ddmm.mmmm` tedy stupně a minuty. Kvůli úspoře dat se přijatý formát uloží jako 4 bytové celé číslo s pevnou desetinnou čárkou se 4 místy. Další úsporu dat přinese označení severní nebo jižní zeměpisné šířky podle znaménka. Jde-li o severní bude znaménko kladné naopak záporné. Stejně funguje zpracování zeměpisné délky přijaté ve formátu `dddmm.mmmm`. Východní zeměpisná délka má znaménko kladné a západní záporné. Čas je přijat ve formátu `hhmmss.sss` a ukládá se jako řetězec znaků, kdy se vezme část hodin, minut a sekund. Všechny tyto informace se nakonec uloží do kruhového bufferu.

Intenzita pohybu

Modul je postavený nad modulem akcelerometru. Používá jediné dvě funkce ze struktury `Module_t`. Funkci `read` a funkci `sleep`. Zpracování dat obstarává akcelerometr a zároveň ukládá data do bufferu pro modul. Samotný modul pouze čte data z bufferu, která vrací. Buffer je vlastně tabulka, která obsahuje pět hodnot určujících intenzitu pohybu:

1. klidový stav
2. téměř klidová pohybová aktivita

3. mírná pohybová aktivita (např. chůze)
4. střední pohybová aktivita (např. rychlá chůze)
5. velká pohybová aktivita (např. běh)

Při volání funkce `read` se vrací data a v tabulce se hodnoty nastaví na nulu. Když jsou všechna data v bufferu nula, tak se nic nepřemazává. Návrh je takový, že se získá tabulka obsahují intenzity aktivit. Buď od spuštění zařízení, nebo od posledního vyžádání dat z tohoto modulu. Pomůže to získat lepší přehled o intenzitě vykonávaných aktivit za kratší časové období. Druhým efektem je, že se může použít méně místa v paměti pro uložení dat a nebude docházet k přetečení.

4.2 Aplikace sběrného zařízení

Byla vytvořena mobilní aplikace jako testovací nástroj pro ověření správné funkčnosti firmwaru měřicího zařízení. Aplikace je vytvořena pro platformu Android. Aplikace je naprogramována pro Android verzi 10.0 a kompatibilita je zaručena až na Android verzi 6.0.

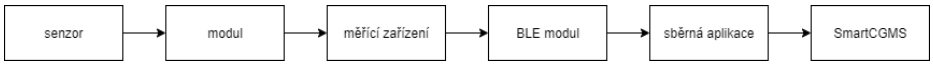
Aplikace obsahuje pouze základní funkčnost pro komunikaci s měřicím zařízením a odesíláním dat do SmartCGMS. Pro tento účel aplikace obsahuje tři základní pohledy. V prvním *úvodním pohledu* se provádí skenování dostupných BLE modulů. Jakmile jsou zařízení naskenována, může se uživatel k jednomu z nich připojit pomocí tlačítka `connect` u nalezeného zařízení.

Po připojení vnitřní mechanismus aplikace zařídí, že se odešle zahajovací zpráva (viz sekce 3.2.2) a měřicí zařízení začne zpracovávat data. Když akce proběhne, je možné začít získávat z měřicího zařízení data. K tomu jsou určena dvě tlačítka. Tlačítko `AUTO` slouží pro automatické získávání dat a tlačítko `DATA` pro vyžádání pouze jednoho kusu dat. Ať už je zapnutý automatický režim, nebo je vyžádán jenom jeden kus dat, vždy se vyšle požadavek na data ze všech senzorů. Automatický režim funguje tak, že se vyšle první zpráva s požadavkem na data a jakmile přijde odpověď, posílá se další požadavek. Pokud by odpověď nepřišla, je po vypršení časového limitu odeslána zpráva s požadavkem na data znovu. Současně s přijímáním dat se data ukládají do souboru a odesílají do filtru SmartCGMS (viz sekce 4.3), pokud je navázáno spojení.

Když je mobilní aplikace připojena k měřicímu zařízení, je možné přepínat z *okna pro zobrazení dat* do *okna nastavení*. Okna se přepínají pomocí menu aplikace. V okně nastavení může uživatel dělat dvě možná nastavení měřicího zařízení. Nastavení doby uspání hlavní smyčky nastavením hodnoty intervalu a zmáčknutím tlačítka `SET`. Nastavení modulů měřicího zařízení, kde se nastavuje interval zpracování dat z modulů a uspání modulu. Proto je potřeba nejdříve získat seznam informací o modulech, a to se zajistí stisknutím tlačítka `GET INFO`. Když jsou získána data o modulech, vypíšou do okna, a je možné je upravit a poslat na měřicí zařízení tlačítkem `SET` v pravém dolním rohu okna.

4.3 Integrace se SmartCGMS

Aby bylo možné dostat data do SmartCGMS, byl vytvořen filtr pro tento software. Jak vypadá průběh získání dat až po uložení dat do SmartCGMS je znázorněno na obrázku 4.2. Filtr je naimplementován tak, že funguje jako TCP server pro přijímání dat, která odpovídají navrženému protokolu (viz sekce 3.5). V souboru `netreciever.cpp` je metoda `Do_Configure`. Ve zmíněné funkci se přebírají parametry s konfigurací. V konfiguračních parametrech je IP adresa a port, na kterých bude server komunikovat. Z parametrů se vytvoří ve funkci serverový socket a vlákno pro obstarání komunikace. Server dokáže obstarávat komunikaci pouze s jedním klientem. Zpracování komunikace probíhá ve funkci `Thread_Func`. Funkce obstarává navázání spojení, ukončení spojení a příjem dat. Vstupním a zároveň jedním z výstupních bodů filtru je funkce `Do_Execute`. Fukce přijímá událost od předchozího filtru a událost může zpracovat nebo poslat na další filtr v pořadí. V případě přijetí události s kódem `Shut_Down`, dojde k zavření serverovému socketu a ukončení činnosti vlákna pro zpracování příchozí a odchozí komunikace.



Obrázek 4.2: Proces získání dat ze senzoru až po odeslání do SmartCGMS

4.3.1 Protokol

Implementace protokolu ve filtru zajišťuje vytváření událostí (viz sekce 1.2). Je implementováno zpracování jediné zprávy protokolu (viz sekce 3.5.2). Ve

zprávě může být přeneseno až několik stovek hodnot získaných ze senzorů. Z každé hodnoty se vytvoří nová událost a posílá se na další filtr v pořadí. Implementace umí vytvářet pět typů signálů, kdy signál je uveden jako parametr v události, a to teplotu kůže, teplotu okolí, zeměpisnou šířku, zeměpisnou délku a intenzitu pohybu.

5 Testování

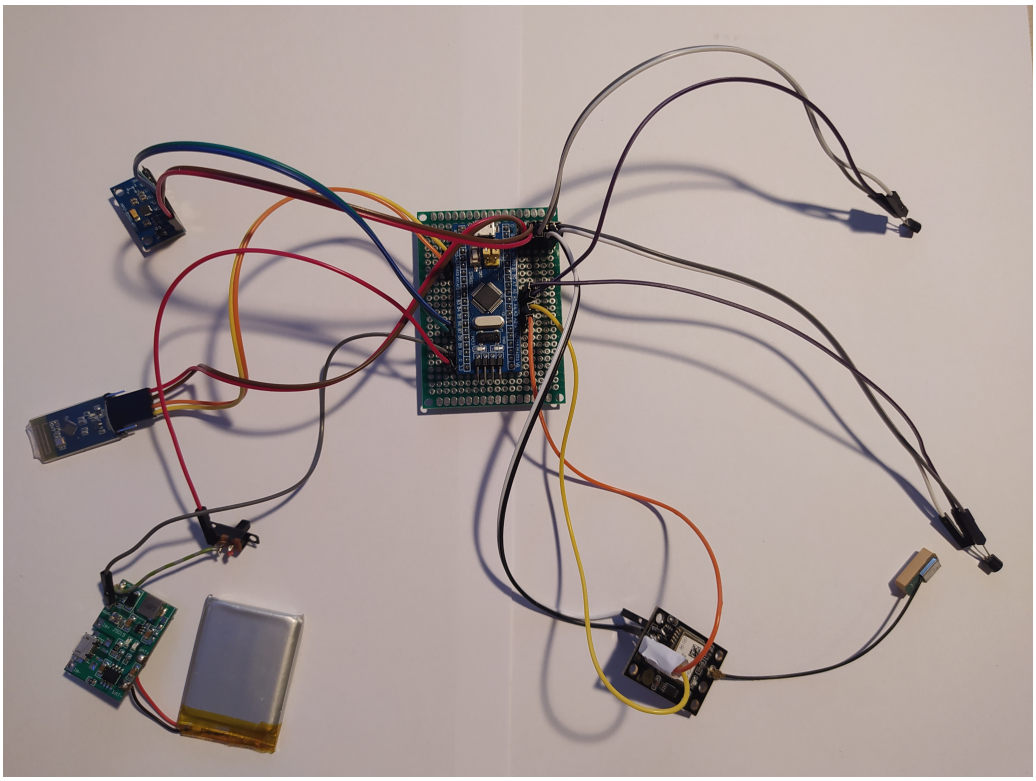
V této kapitole budou prezentovány výsledky testování měřicího zařízení. Funkčnost měřicího zařízení bude otestována vytvořenou sběrnou aplikací. Hlavním účelem testu bude zjistit přesnost naměřených veličin měřicím zařízením. Hodnoty naměřených veličin ze senzorů připojeným k měřicímu zařízení budou srovnány s naměřenými hodnotami z konvenčních zařízeních. V poslední řadě budou výsledky krátce diskutovány.

5.1 Celková funkcionalita

Než se bylo možné dostat k funkčnosti software, bylo potřeba ověřit hardware měřicího zařízení, kdy výsledné sestavení je vyobrazeno na obrázku 5.1. Jak bylo napsáno výše, měřicí zařízení má k sobě připojeno několik dalších elektronických součástek připojených pomocí vodičů. Bylo potřeba ověřit, jestli je funkční napájení baterií a zda je kontakt mezi všemi propojeními elektrických součástek.

Kontakt a napájení se ověřoval pomocí multimetru `PROSKIT MT-1210` . Při měření multimetrem nebyl zjištěn problém mezi propojeními. Je zde ale problém, že vodiče propojující součástky nejsou dostatečně odolné a je snadné je porušit. Při možném budoucím vývoji by bylo vhodné opatřit měřicímu zařízení a jeho součástkám odolnou strukturu, aby byly vodivé dráhy chráněny. Bez této struktury je i poněkud nepraktické zařízení umisťovat na lidské tělo.

Co se funkčnosti napájení týče, zde vyvstal problém s měničem napětí. Při testování se poškodil šroub regulující úroveň napětí. Kvůli tomu nebylo možné nastavit nižší napětí na výstupu DC měniče na předepsané vstupní napětí desky $3.3V$. Hodnota napětí se pohybovala kolem $4.1V$ a to způsobo-



Obrázek 5.1: Výsledná podoba měřicího zařízení

valo problémy s ADC převodníkem. Kvůli tomuto problému byly provedeny testy měření teploty, kde se využívá ADC, se zapojeným měřicím zařízením v elektrické síti. ADC převodník potřebuje pro správnou funkci předepsanou hodnotu napětí ($3.3V$), jinak naměřené hodnoty nejsou přesné. V případě dalšího využití měřicího zařízení mimo tuto práci by se měl DC měnič vyměnit, protože hodnota napětí musí být v rozsahu $2.2V$ až $4.0V$, jak udává výrobce [26]. V případě vyšší hodnoty napětí než udává výrobce, může dojít k poškození desky.

Při testování software bylo potřeba se zaměřit na funkcionalitu měřicího zařízení a přenosu dat do SmartCGMS. Funkcionalita měřicího zařízení byla ověřována pomocí aplikace sběrného zařízení, která byla pro tento účel vytvořena. Testovalo se, zda měřicí zařízení reaguje správně na zprávy protokolu (viz sekce 4.1.2) posílané ze sběrného zařízení. Měřicí zařízení funguje dle návrhu a na všechny implementované zprávy reaguje korektně. Jediným problémem je, že občas nepřijde z měřicího zařízení odpověď na zprávu. Tento jev byl zjištěn na základě pozorování počtu úspěšně přijatých datových zpráv (viz sekce 3.2.6) a na základě odesílání zpráv s požadavkem na data (viz sekce 3.2.3), kdy datová zpráva nedorazila v časovém limitu. Přitom

byla určena míra spolehlivosti přenosu 86 %, na základě vzorku 100 zpráv, kdy jich dorazilo 86 v pořádku. Tento problém může být způsobený krátkým časovým intervalem pro zpracování zprávy na měřicím zařízení. Tedy je možné, že se na měřicí zařízení posílají další zprávy, přestože měřicí zařízení zpravovává ještě předchozí. Jelikož se nepodařilo chybu v implemetaci identifikovat, připadá v úvahu i možnost, že je chyba ve firmware použitého BLE modulu.

Ověření funkčnosti implementovaného filtru do SmartCGMS proběhlo taktéž pomocí aplikace sběrného zařízení. Testovaly se dva režimy odesílání dat do SmartCGMS. V prvním testu se odesílaly zprávy průběžně – jakmile byla přijata data z měřicího zařízení, byla odeslána do SmartCGMS. V druhém testu sběrné zařízení získávalo přibližně 10 minut data a poté byla odeslaná. V obou případech byly všechny naměřené hodnoty přeneseny ve správném pořadí.

5.2 Test spotřeby energie

U zařízení byl kladen důraz na co nejmenší spotřebu energie a tento test se zaměřuje na zhodnocení tohoto kritéria. K ověření byl použit multimetr a byl připojen mezi záporný pól výstupu DC měniče a záporný pól měřicího zařízení.

Byl proveden test spotřeby elektrické energie, kdy měřicí zařízení mělo zapnuté všechny senzory a interval zpracování dat byl 100ms. Při tomto testu byla naměřena hodnota odebíraného proudu $85mA$. V případě běžného použití se počítá s baterií o kapacitě $1500mAh$. V tomto případě by zařízení na této baterii vydrželo pracovat až 17.6 hodin.

Tato spotřeba je na standardy dnešních zařízení relativně vysoká, ale vzhledem k použitým senzorům a jejich konfiguracím je opodstatněná. Spotřeba by se dala dále snižovat například vypínáním některých senzorů na dobu, kdy není potřeba tato data číst. Například pokud je celé zařízení v klidu dle detektoru pohybu, nemusí být nutné soustavně zjišťovat polohu GPS modulem. Momentálně lze vypínání modulů docílit ručně prostřednictvím sběrné aplikace.

5.3 Přesnost měření teploty prostředí

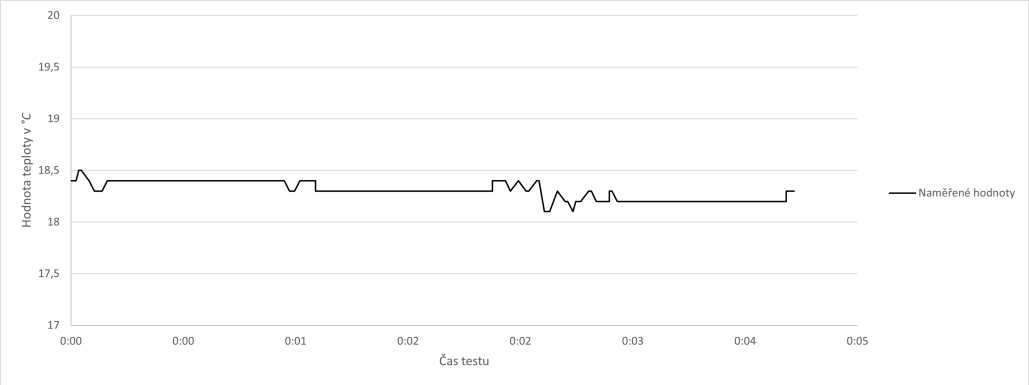
Hodnoty naměřené senzorem se porovnávaly vůči teploměru `Auriol Z31482B`, který měřil jak teplotu bytu, tak venkovní teplotu. První dva testovací případy se zaměřují na přesnost měření, pokud bude senzor vystaven po delší dobu stálé teplotě. V prvním případě se měřila teplota bytu a v druhém venkovní teplota. Poslední případ se zabývá tím, jak dlouho senzoru trvá získat přibližnou hodnotu teploty okolí při změně prostředí. Při testu se nejdříve nechal senzor stabilizovat na venkovní teplotu a poté byl senzor vystaven vnitřní teplotě.

První test, jehož výsledky jsou na obrázku 5.2, testoval přesnost naměřených hodnot ve vnitřním prostředí. Testovalo se vůči teplotě 20.1°C, která se na referenčním teploměru po dobu testu neměnila. Z výsledků je vidět že naměřené hodnoty se liší pouze o 1.5°C. Souvislosti s touto nepřesností jsou uvedeny v dalších odstavcích.

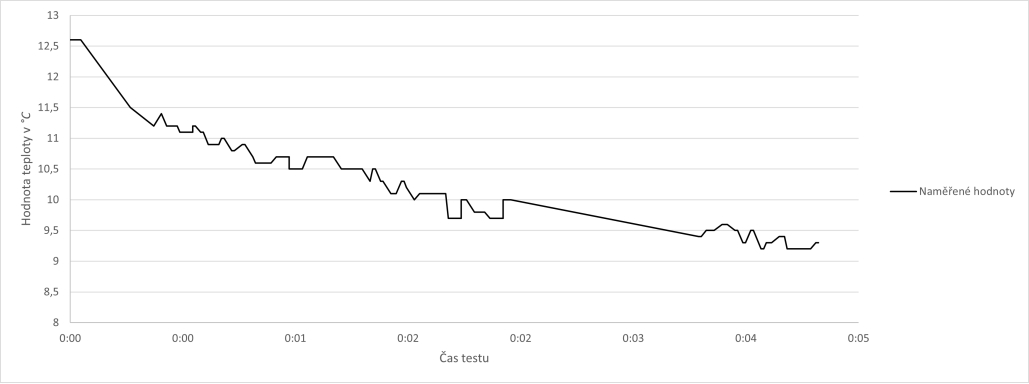
V druhém testu, kdy se testovala přesnost senzoru ve venkovním prostředí, jsou výsledky uvedené na obrázku 5.3. Referenční venkovní teplota byla 11.3°C, a postupně se měnila na hodnotu 11°C. Zde je závěr podobný jako u předchozího testu. Výsledná hodnota se liší rovněž o 1.5°C.

Nepřesnost mezi naměřenou hodnotou referenčním teploměrem a senzorem TMP36, může být zapříčiněna buď součtem chyb měření senzoru a teploměru, nebo omezené přesnosti ADC převodníku. Přesto, i s chybou která se pohybuje okolo 1.5°C, je senzor pro účely měření okolní teploty použitelný.

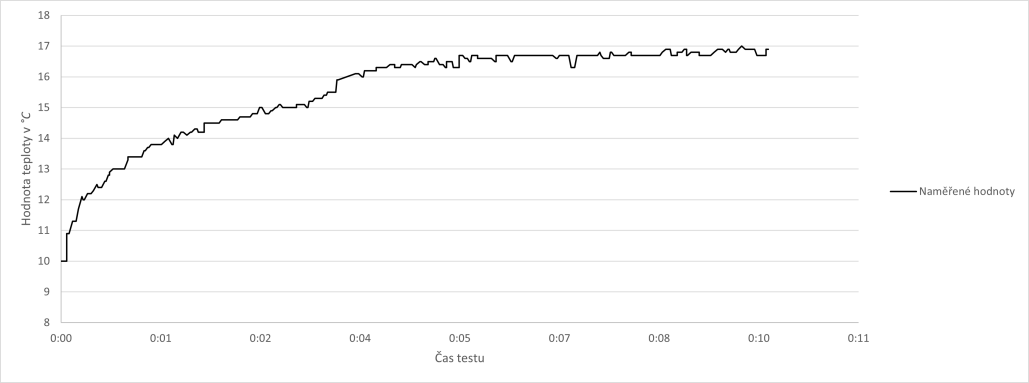
V posledním testu se zjišťovala doba trvání ustálení vracené hodnoty ze senzoru při změně prostředí. Při testu byla na referenčním teploměru naměřena venkovní teplota 11.4°C a vnitřní teplota 18°C. Potřebnou dobu pro ustálení hodnot je možno vyčíst z grafu na obrázku 5.4 a ta je přibližně 4 minuty. Podobně jak v předchozích testech i zde se počáteční teplota lišila přibližně o 1.5°C. Vnitřní teplota byla při tomto testu nižší něž v prvním testu z důvodu větrání v místnosti.



Obrázek 5.2: Test stálosti teploty uvnitř budovy



Obrázek 5.3: Test stálosti venkovní teploty



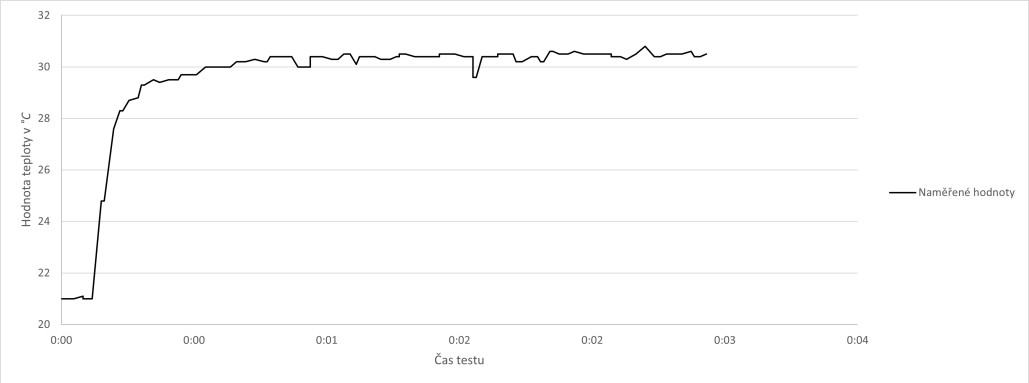
Obrázek 5.4: Test přizpůsobení teploty při změně prostředí

5.4 Přesnost měření tělesné teploty

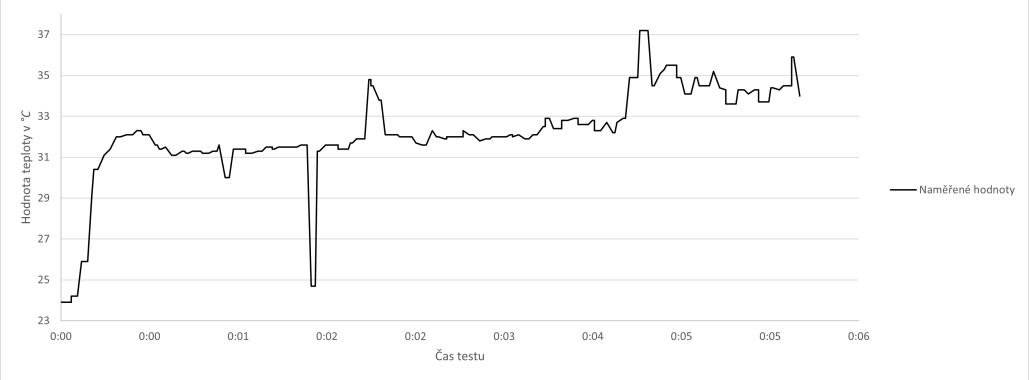
Měřicí zařízení je potřeba mít na těle umístěné, tak aby neomezovalo běžné činnosti. Pro test se zvolilo měření ze hřbetu ruky. Pro ověření výsledků se provedlo měření také běžně dostupným teploměr pro měření teploty těla z podpaží.

Jako první bylo provedeno měření běžně dostupným teploměrem *Hartmann Thermoval*. Byla provedena měření tímto teploměrem, kdy výsledky byly 36.2°C, 36.2°C a 36.1°C. Oproti tomu byly naměřené výsledky viz obrázek 5.5 senzorem TMP36 ze hřbetu ruky. Senzor zpočátku vracel hodnoty teploty, které se postupně blížily očekáváné teplotě, ale přibližně po 40 sekundách testu se hodnoty ustálily. Od naměřené hodnoty běžně dostupným teploměrem se hodnoty liší přibližně o 6°C. Z tohoto výsledku vyvstala otázka, jestli je hřbet ruky vhodné místo k měření teploty těla tímto senzorem, nebo senzor není vhodný k tomuto měření.

Bylo provedeno ještě jedno měření teploty z čela, kdy výsledky jsou uvedené na obrázku 5.6. Při tomto měření už byly zaznamenány lepší výsledky, kdy se teplota po ustálení lišila o přibližně 3°C a pár hodnot ke konci měření se lišilo pouze o 1°C. I přesto, že bylo druhé měření úspěšnější, nezbývá než konstatovat, že senzor TMP36 se na měření teploty těla nehodí. Jak ukázalo měření teploty těla ze hřbetu ruky, kde by mohl být senzor pohodlně připevněný, je velice nepřesné. Oproti tomu umístěním na čele už by šlo získat přesnější hodnoty, ale stále se lehce lišící od očekávaného výsledku, a navíc by bylo umístění velice nepraktické. Pro měření teploty těla by v tomto případě stálo za zvážení využití bezdotykového teploměru, který by se přikládal k čelu za určitý časový interval. Tento senzor by nemohl být napevno umístěn, ale musel by se umisťovat například před čelo, kde byly výsledky přesnější. K měření teploty by pak měřicí zařízení mohlo vyzvat na základě zvukového podnětu. Samotné měření by se spouštělo například stisknutím tlačítka, které by se pro tyto účely na měřicí zařízení přidalo.



Obrázek 5.5: Test měření teploty těla na hřbetu ruky



Obrázek 5.6: Test měření teploty těla na čele

5.5 Ověření přesnoti pohybového modulu

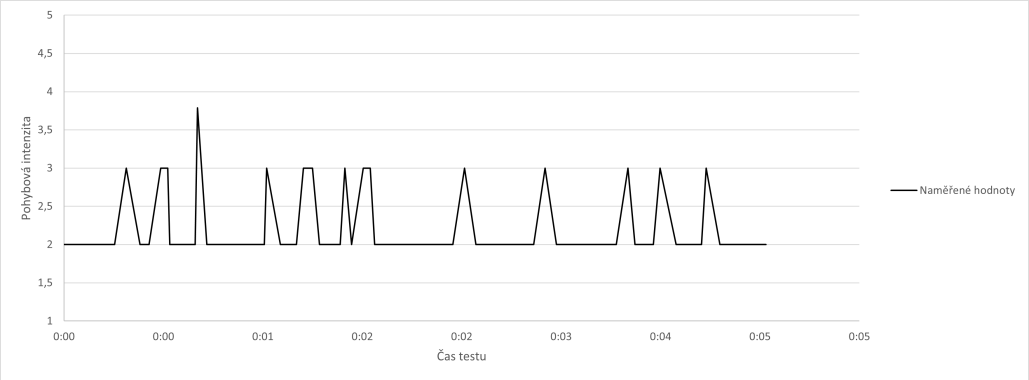
Byly provedeny tři testy. V prvním testu se sleduje správnost vrácených hodnot, když je zařízení v klidovém stavu. V druhém testu je se zařízením prováděna mírná pohybová aktivita (chůze). V posledním testu je zařízení vystavené střední až velké pohybové aktivitě (běh).

Při prvním testu bylo zařízení položené na stole po dobu 5 min. V průběhu testu se sledovalo, zda modul správně registruje nulovou intenzitu pohybu. Jak je vidět na obrázku 5.9, modul vrací hodnoty dle očekávání pokud je v klidu. Vyšší intenzita pohybu než nulová se v grafu na obrázku 5.9 objevila z důvodu menšího otřesu stolu při testu.

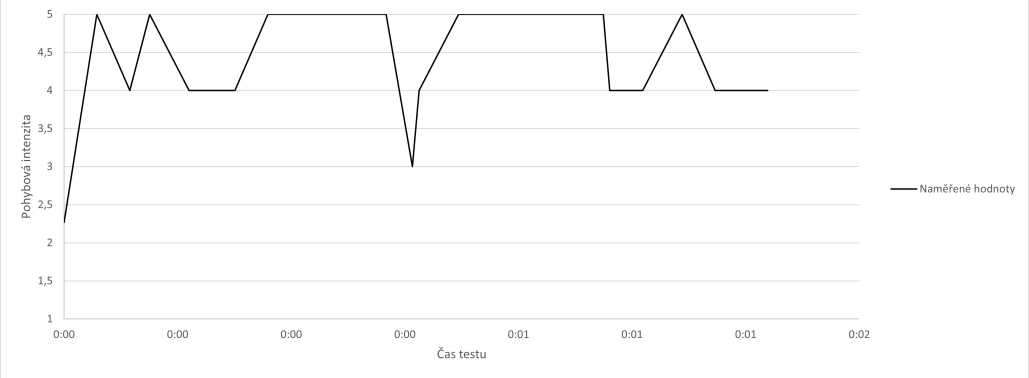
V druhém testu bylo za cíl otestovat, zda při normální chůzi modul vrací tomu odpovídající pohybovou intenzitu. Test byl prováděn na trase vyobrazené na obrázku 5.10 a rychlost pohybu se pohybovala v rozmezí 2 až 5 km/h. Ze získaných výsledků, které jsou na obrázku 5.7 je vidět, že pohybová intenzita byla mezi 2. až 3. stupněm, což přibližně odpovídá stanové stupnici (viz sekce 4.1.3). Rychlost chůze nebyla místy příliš velká, což odpovídá druhému stupni.

V posledním testu byl vyzkoušen běh se zařízením, kdy se rychlost pohybovala v rozmezí 8 až 10 km/h. Test probíhal na stejné trase jako test zmíněný v předchozím odstavci. Pohybová intenzita se pohybovala mezi 4. až 5. stupněm, jak je vidět na obrázku 5.8. I v tomto testu hodnoty odpovídají přibližně stanovené stupnici.

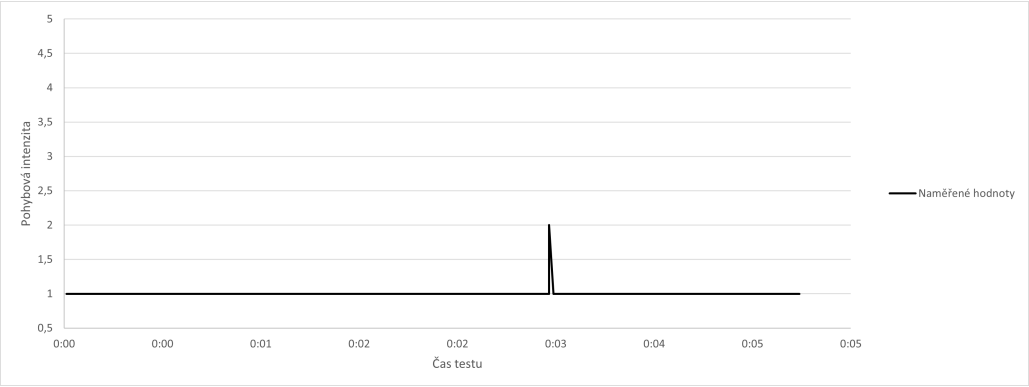
Na základě výsledku lze říct, že pohybový senzor funguje dle očekávání, pokud je prováděna fyzická aktivita. Detekce fyzické aktivity nebyla předmětem této práce a implementovaný algoritmus slouží pouze pro ověření funkčnosti implementace modulu akcelerometru. Předmětem dalšího rozšíření práce může být lepší detekce intenzity fyzické aktivity.



Obrázek 5.7: Test pohybového modulu při chůzi



Obrázek 5.8: Test pohybového modulu při běhu



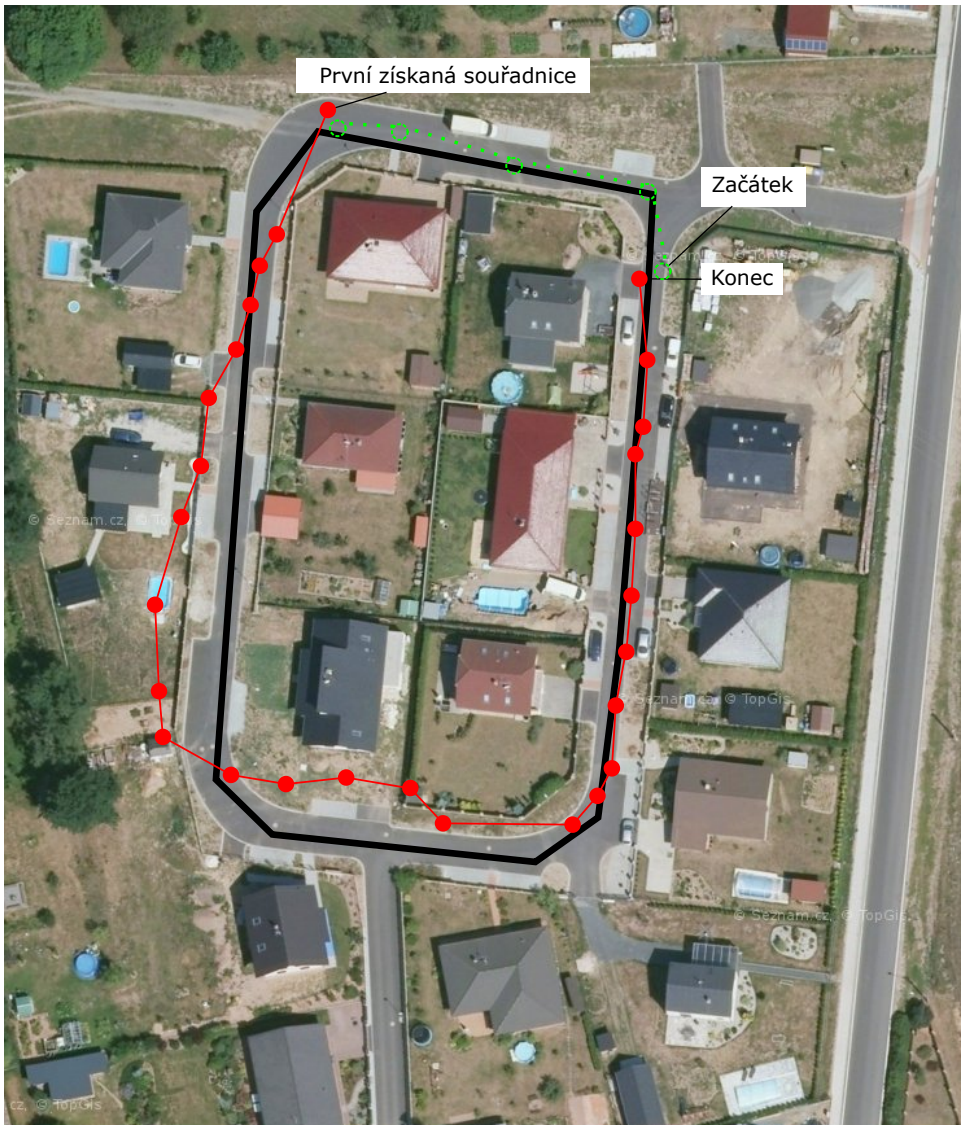
Obrázek 5.9: Test pohybového modulu v klidu

5.6 Ověření přesnosti GPS modulu

Testování přesnosti vrácených GPS souřadnic proběhlo ve venkovním prostředí na trase vyobrazené na obrázku 5.10. V tomto testu byl testován pouze GPS modul. Plná celá čára představuje reálnou trasu, na které byl prováděn test. Červené body spojené červenou čarou představují data získaná modulem. Zelené body spojené zelenou čárkovanou čarou představují úsek od zapnutí měřicího zařízení po získání první GPS souřadnice. Celková trasa měří přibližně 300 m. Při testu se pohyboval senzor rychlostí přibližně v rozmezí 2 km/h až 4 km/h.

Doba získání první souřadnice byla přibližně 30 sekund. To odpovídá zeleně vyznačenému úseku, který měří necelých 40 metrů. Po tomto úseku se začaly získávat souřadnice pravidelně. Jak je vidět na obrázku zpočátku byla přesnost do 10 metrů od reálné trasy. Ke konci testu se přesnost zvýšila řádově na jednotky metrů od reálné trasy. Získání lepší přesnosti trvalo přibližně 3 minuty.

Na základě výsledků testu je přesnost dostačující. Musí se ale brát ohled na to, že je potřeba několik jednotek minut počkat, než modul začne získávat první souřadnice a než se zvýší přesnost. Je třeba brát v úvahu, že senzor má pouze venkovní využití (viz sekce 4.1.3).



Obrázek 5.10: Test GPS modulu

6 Závěr

Cílem této práce bylo analyzovat veličiny ovlivňující průběh nemoci diabetes mellitus, vybrat vhodné senzory k jejich měření a vybrat vhodné zařízení pro sběr naměřených hodnot ze senzorů. Dalším krokem byl návrh a implementace měřicího zařízení, aplikace sběrného zařízení a vytvoření knihovny pro SmartCGMS. Posledním krokem bylo všechny části práce otestovat a zhodnotit funkčnost řešení.

Zadání práce bylo splněno v celém rozsahu. Bylo vytvořeno měřicí zařízení a implementován firmware pro něj. Firmware zajišťuje získávání dat ze senzorů a jejich zpracování do vhodné podoby. Z měřicího zařízení se data odesílají navrženým protokolem do sběrného zařízení. Nakonec byl implementován modul do SmartCGMS, aby bylo možné zajistit předání dat do zmíněné aplikace. Modul byl implementován v podobě filtru do SmartCGMS. Filtr implementuje jednoduchý protokol (viz sekce 3.5), pomocí kterého se přenáší data z aplikace sběrného zařízení.

Při dalším vývoji měřicího zařízení by bylo zajímavé ho rozšířit o senzor pro měření tepové frekvence, kdy tepová frekvence vyšla jako zajímavá veličina v analýze (viz sekce 2.1.4). Dále by bylo dobré vytvořit pro měřicí zařízení vhodný obal, aby šlo vhodně umístit na lidské tělo.

Navázáním na tuto práci by mohlo být vyhodnocování naměřených dat v souvislosti s nemocí diabetes mellitus. Při zpracování dat z měřicího zařízení by bylo možné částečně předcházet akutním stavům spojeným s diabetem jako je *hypoglykémie* nebo *hyperglykémie*.

Je zde prostor pro další zlepšení měřicího zařízení. Použitý senzor TMP36 se ukázal jako nepříliš vhodný pro měření tělesné teploty (viz sekce 5.4). Pro zlepšení přesnosti měření by bylo vhodné tento senzor vyměnit za kvalitnější senzor, umožňující přesnější měření teploty těla. Při pozdější fázi tvorby kódu firmware byl problém s místem na flash paměti zapříčiněno využíváním

knihovny k výpisu dat na komunikační port. Knihovna zabírala přibližně třetinu místa na flash paměti. Kdyby se mělo měřicí zařízení dále vyvíjet, bylo by vhodné zvolit jiný MCU s větší kapacitou flash paměti.

Literatura

[1] Apelqvist, J. et al. International consensus and practical guidelines on the management and the prevention of the diabetic foot. *Diabetes/-metabolism research and reviews*. 2000, 16, S1, s. S84–S92.

[2] *ATmega328P* [online]. Atmel Corporation. [cit. 28.4.2021]. Dostupné z: `https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf`.

[3] Asaduzzaman, A. – Chidella, K. K. – Mridha, M. F. A Time and Energy Efficient Parking System Using ZigBee Communication Protocol. *ResearchGate*. 5 2015.

[4] Barragán, H. *Wiring* [online]. [cit. 28.4.2021]. Dostupné z: `http://wiring.org.co/`.

[5] Barry, P. Abstract syntax notation-one (ASN. 1). In *IEE Tutorial Colloquium on Formal Methods and Notations Applicable to Telecommunications*, s. 2–1. IET, 1992.

[6] Beningo, J. *Decreasing energy consumption using direct memory access* [online]. 2018. [cit. 28.4.2021]. Dostupné z: `https://www.embedded.com/decreasing-energy-consumption-using-direct-memory-access/`.

[7] *Bluetooth SIG* [online]. [cit. 28.4.2021]. Dostupné z: `https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=286439l`.

[8] *Bluetooth Market Update 2019* [online]. Bluetooth SIG, Inc., 2019. [cit. 28.4.2021]. Dostupné z: `https://www.bluetooth.com/wp-content/uploads/2018/04/2019-Bluetooth-Market-Update.pdf`.

[9] Castaneda, D. et al. A review on wearable photoplethysmography sensors and their potential future applications in health care. *International journal of biosensors & bioelectronics*. 2018, 4, 4, s. 195.

[10] Daidone, R. – Dini, G. – Anastasi, G. On evaluating the performance impact of the IEEE 802.15. 4 security sub-layer. *Computer Communications*. 2014, 47, s. 65–76.

[11] *ESP32WROOM32* [online]. Espressif Systems (Shanghai) Co., Ltd. [cit. 28.4.2021]. Dostupné z: `https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf`.

[12] Gajbhiye, S. et al. Bluetooth secure simple pairing with enhanced security level. *Journal of information security and applications*. 2019, 44, s. 170–183.

[13] Ganong, W. F. *Přehled lékařské fyziologie*. Galen, 2005. ISBN 978-80-85787-36-8.

[14] Hall, J. E. – Hall, M. E. *Guyton and Hall textbook of medical physiology e-Book*. Elsevier Health Sciences, 2020.

[15] Johnson, A. *Wireless Concepts* [online]. Cisco Press, 2020. [cit. 28.4.2021]. Dostupné z: `https://www.ciscopress.com/articles/article.asp?p=2999384&seqNum=3`.

[16] Josang, A. – Miralabe, L. – Dallot, L. It's not a bug, it'sa feature: 25 years of mobile network insecurity. In *ECCWS2015-Proceedings of the 14th European Conference on Cyber Warfare and Security 2015: ECCWS 2015*, s. 129. Academic Conferences Limited, 2015.

[17] Karen, I. – Svačina, . *Diabetes mellitus : doporučené diagnostické a terapeutické postupy pro všeobecné praktické lékaře*. Společnost všeobecného lékařství ČLS JEP, 2020. ISBN 978-80-88280-16-3.

[18] Kazeem, O. O. – Akintade, O. O. – Kehinde, L. O. Comparative Study of Communication Interfaces for Sensors and Actuators in the Cloud of Internet of Things. *Int. J. Internet Things*. 2017, 6, 1, s. 9–13.

[19] Koutny, T. – Ubl, M. Parallel software architecture for the next generation of glucose monitoring. *Procedia Computer Science*. 2018, 141, s. 279–286.

[20] Leach, P. – Mealling, M. – Salz, R. A universally unique identifier (uuid) urn namespace. RFC 4122, RFC Editor, 2005. Dostupné z: https://www.hjp.at/doc/rfc/rfc4122.html.

[21] Petrova, M. et al. Performance study of IEEE 802.15. 4 using measurements and simulations. In *IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006.*, 1, s. 487–492. IEEE, 2006.

[22] *u-blox 6 Receiver Description* [online]. u-blox AG, 2013. [cit. 27.04.2021]. Dostupné z: https://www.u-blox.com/en/ubx-viewer/view/u-blox6_ReceiverDescrProtSpec_(GPS.G6-SW-10018)_Public?url=https%3A%2F%2Fwww.u-blox.com%2Fsites%2Fdefault%2Ffiles%2Fproducts%2Fdocuments%2Fu-blox6_ReceiverDescrProtSpec_%2528GPS.G6-SW-10018%2529_Public.pdf.

[23] Rosolová, H. – Pelikánová, T. – Mořovská, Z. Doporučené postupy ESC týkající se diabetu, prediabetu a kardiovaskulárních onemocnění, vytvořené ve spolupráci s EASD. *Elsevier Urban & Partner Sp. z o.o. on behalf of the Czech Society of Cardiology*. 2014, s. 229–246.

[24] Roth, G. Bluetooth Wireless Technology, 2012. Dostupné z: http://large.stanford.edu/courses/2012/ph250/roth1/l.

[25] Senn, M. *Data o diabetu v ČR*, 2015 (accessed February 3, 2014). http://diabetickaasociace.cz/co-je-diabetes/data-o-diabetu-v-cr/.

[26] *STM32F103x8* [online]. STMicroelectronics, Inc. [cit. 28.4.2021]. Dostupné z: https://www.st.com/resource/en/datasheet/stm32f103c8.pdf.

[27] Surwit, R. S. – Schneider, M. S. – Feinglos, M. N. Stress and Diabetes Mellitus. *Diabetes Care*. 10 1992, 15, s. 229–246.

[28] *CC2640 SimpleLink Bluetooth Wireless MCU* [online]. Texas Instruments , Inc. [cit. 28.4.2021]. Dostupné z: https://www.ti.com/lit/ds/symlink/cc2640.pdf?ts=1619647796177&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCC2640.

[29] Ubl, M. – Koutny, T. SmartCGMS as an environment for an insulin-pump development with fda-accepted in-silico pre-clinical trials. *Procedia Computer Science*. 2019, 160, s. 322–329.

[30] Vokurka, M. – Hugo, J. *Velký lékařský slovník*. Maxdorf s. r. o., 2004. ISBN 80-7345-037-2.

[31] Vondruška, V. – Barták, K. *Pohybová aktivita ve zdraví a v nemoci*. Klinika tělovýchovného lékařství FN a LFUK, 1999. ISBN 80-238-4536-5.

[32] Wikipedia contributors. Stress (biology) — Wikipedia, The Free Encyclopedia, 2020. Dostupné z: `https://en.wikipedia.org/w/index.php?title=Stress_(biology)&oldid=982122563`. [Online; accessed 22-October-2020].

[33] *Wireless and Network Security Integration Solution Design Guide* [online]. Cisco Systems, Inc. [cit. 28.4.2021]. Dostupné z: `https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/secwlandg20/sw2dg/ch3_2_SPMb.html`.

Seznam zkratek

ACK Acknowledgement - Kladné potvrzení příjmu dat

ADC Analog To Digital - Převod analogového signálu na digitální

ASCII American Standard Code for Information Interchange - Standard pro kódování znaků

BLE Bluetooth Low Energy - Bezdrátová technologie vycházející z Bluetooth vyznačující se nízkou spotřebou energie

CGMS Continuous Glucose Monitoring System - Systém monitorování hladiny glukózy v krvi

CSV Microcontroller Unit - Mikropočítač sestávající se z procesoru, paměti, sběrnic a dalšího hardwarového vybavení

DMA Direct Memory Access - Přenášení dat mezi vstupně-výstupními zařízeními a operační pamětí bez zásahu procesoru

IDE Integrated Development Environment - Vývojové prostředí usnadňující vývoj software

IoT Internet of Things - Síť zařízení, vybavena senzory a software, schopných si navzájem vyměňovat data

I2C Inter-Integrated Circuit - Seriová sběrnice využívaná pro připojení nízkorychlostních periferií

MCU Microcontroller Unit - Mikropočítač sestávající se z procesoru, paměti, sběrnic a dalšího hardwarového vybavení

NACK Negative-Acknowledgement - Záporné potvrzení příjmu dat

PAN Personal Area Netowrk - Síť propojující zařízení v bezprostřední blízkosti uživatele

PPG Photoplethysmogram - Metoda pro získání hodnoty průtoku krve

SPI Serial Peripheral Interface - Seriová sběrnice využívaná pro připojení nízkorychlostních periferií

UART Universal Asynchronous Receiver-Transmitter - Seriová sběrnice pro asynchronní přenos dat

Seznam obrázků

2.1 Hierarchie GATT [7] 21

2.2 Senzor MPU-9250, Zdroj: https://dratek.cz/arduino/1330-9dof-gyroskop-akcelerometr-magnetometr-mpu-9250-spi-iic-modul-pro-arduino.html 33

2.3 GPS modul, Zdroj: https://www.elecdesignworks.com/shop/ublox-neo-6m-gps-module-red-mini-indoor-antenna/ . 34

2.4 Senzor teploty TMP36. Zdroj: https://www.robotshop.com/en/temperature-sensor-tmp36.html 35

2.5 Senzor MAX30100, Zdroj: http://www.alselectro.com/max30100-pulse-oximeter-spo2-and-heart-rate-sensor-module.html 36

3.1 Schéma zapojení sběrného zařízení 42

3.2 Stavový diagram průběhu komunikace měřicího a sběrného zařízení 50

4.1 Struktura firmware měřicího zařízení 56

4.2 Proces získání dat ze senzoru až po odeslání do SmartCGMS 67

5.1 Výsledná podoba měřicího zařízení 70

5.2 Test stálosti teploty uvnitř budovy 73

5.3 Test stálosti venkovní teploty 73

5.4 Test přizpůsobení teploty při změně prostředí 74
5.5 Test měření teploty těla na hřbetu ruky 76
5.6 Test měření teploty těla na čele 76
5.7 Test pohybového modulu při chůzi 78
5.8 Test pohybového modulu při běhu 78
5.9 Test pohybového modulu v klidu 79
5.10 Test GPS modulu 81

Seznam tabulek

2.1 Tabulka srovnání bezdrátových technologií 26

2.2 Srovnání základních parametrů desek 30

2.3 Srovnání vybavení desek 30

2.4 Tabulka srovnání platforem 31

3.1 Tabulka intenzity pohybu 39

Obsah přiloženého CD

- **bin** - zkompilované programy
  - firmware - spustitelný kód pro měřicí zařízení (platforma STM32)
  - mobile - aplikace sběrného zařízení (Android)
  - server - filtr do SmartCGMS (MS Windows)
- **sources** - zdrojové kódy
  - firmware - měřicí zařízení
  - mobile - aplikace sběrného zařízení
  - server - filtr do SmartCGMS
- **thesis** - text bakalářské práce včetně zdrojových souborů LaTeX

Příloha A: Instalace software

A.1 Firmware pro měřicí zařízení

V první řadě je potřeba si obstarat několik věcí, než se začne se samotnou instalací. Je potřeba deska STM32F103C8T6 a ST-LINK/V2.

Propojte desku pomocí vodičů na odpovídající piny ST-LINK/V2 a desky. ST-LINK/V2 se propojí s počítačem pomocí USB kabelu. Nyní je potřeba si na naistalovat program STM32CubeProgrammer dostupný na webové adrese `https://www.st.com/en/development-tools/stm32cubeprog.html`. Spustíme naistalovaný program a v pravém menu vybereme možnost `ST-LINK` a klikneme na `Connect`. Jakmile jsme připojení k desce, přejdeme pomocí levého menu stisknutím tlačítka „Eresing & programming“ do okna pro nahrání firmware. Zde zadáme do okna „File path“ cestu k souboru `firmware.bin` . Další parametry se nemění a stiskneme „Start Programming“. Po této akci by měl být firmware nahrán na desce. Pro správnou funkčnost ještě doporučuji vypnout a zapnout desku, než se s ní začne operovat.

A.2 Aplikace sběrného zařízení

K instalaci aplikace je potřeba si nakopírovat do svého mobilní telefonu dodaný instalační soubor `mobile.apk`. Jakmile je soubor překopírovaný, pomocí průzkumníku souborů v mobilním telefonu nalezneme soubor. Na nalezený soubor klikneme a otevře se dialog, kde klikneme na „Nainstalovat“. Postup se může mírně lišit v závislosti na verzi použitého operačního systému Android.

Problém může nastat, pokud není povolena instalace aplikací, které nepo-

cházejí z Google Play. V tomto případě je potřeba jít do nastavení mobilního telefonu a tam povolit instalaci od třetích stran. Po této akci je potřeba se vrátit a zkusit instalaci znovu. Nyní by měla proběhnout bez problémů. Jakmile se instalace dokončí, měla by se ikona aplikace objevit v menu aplikací. Pro správnou funkci aplikace je dobré zkontrolovat, zda má aplikace oprávnění pro získání polohy. Oprávnění pro polohu je důležité kvůli správné funkčnosti BLE.

Alternativně lze instalace provést přímo z počítače. Je potřeba mít naistalované Android SDK. Potom stačí telefon připojit k počítači a příkazem `adb install mobile.apk` naistalovat aplikaci.

A.3 Přidání filtru do SmartCGMS

Nejprve si stáhneme a nainstaluje software SmartCGMS `https://diabetes.zcu.cz/smartcgms`. Najdeme si adresář, kam se software nainstaloval a v tomto adresáři otevřeme adresář *filter*. Do otevřeného adresáře nakopírujeme soubor `server.dll`. Spustíme aplikaci, kde klikneme na File -> Open vybereme další přiložený soubor *config.ini*. Alternativně stačí přetáhnout *config.ini* na iconu aplikace a aplikace se po této akci spustí.

Příloha B: Přenos dat do sběrného zařízení

B.1 Hlavička protokolu

```
Opcode ::= Byte(
   001 | //start message
   002 | //initial message
   003 | //settings message
   004 | //data message
   005 | //request message
   006 | //stop message
   007 | //ACK message
   008   //NACK message
)

MessageHeader {Opcode:opcodeVal} ::= SEQUENCE{
   opcode           opcodeVal,
   mesLegth         Short,
   mesTimeStamp     Integer
}
```

B.2 Zahajovací zpráva

```
StartMessage ::= SEQUENCE{
   header     MessageHeader{{001}}
}
```

B.3 Zpráva s požadavkem

```
ReqType ::= Byte(
   001 | //data
   002 | //info
)

RequestMessageHeader {ReqType:type} ::= SEQUENCE{
   header     MessageHeader
   reqType    type
}
```

B.3.1 Zpráva s požadavkem o informace

```
RequestInfoMessage ::= SEQUENCE{
   header     RequestMessageHeader{{002}}
}
```

B.3.2 Zpráva s požadavkem o data

```
RequestDataMessage ::= SEQUENCE{
   header     RequestMessageHeader{{001}},
   bufferData Boolean,
   modulesIDs SEQUENCE(SIZE(1..255)) OF Byte
}
```

B.4 Informativní zpráva

```
DevType ::= Byte(
   001 |   //accelometr
   002 |   //gyroscope
   003 |   //temp
   004 |   //pulse
   005 |   //GPS
   006     //Movement sensor
```

```
)

ModuleInfo ::= SEQUNECE {DevType: Type}{
   moduleID            BYTE(0..255),
   devType             Type,
   currentReadInterval Short(1..3600),
   minReadInterval     Short(1..3600),
   maxReadInterval     Short(1..3600),
   sleep               BOOLEAN,
   descLen             BYTE(0..16),
   desc                IA5String(Size(0..16)),
}

InfoMessageHeader ::= SEQUENCE {
   header     MessageHeader{{002}},
   modules    SEQUENCE(SIZE(1..255)) OF ModuleInfo
}
```

B.5 Zpráva nastavení

```
SettingsMessageHeader ::= SEQUENCE {ReqType: type} {
   header     MessageHeader{{003}}
   requestType type
}
```

B.5.1 Zpráva nastavení modulů

```
ModuleSettings ::= SEQUENCE {
   moduleID      Byte(0..255),
   readInterval  Short(0..3600),
   sleep         Boolean
}

ModulesSettingsMessage ::= SEQUENCE{
   header     SettingsMessageHeader{{001}}
   modules    SEQUENCE(SIZE(1..255)) of ModuleSettings
}
```

B.5.2 Zpráva nastavení hlavní smyčky

```
MainLoopSettingsMessage ::= SEQUENCE{
   header     SettingsMessageHeader{{002}}
   value      Short(50..1000)
}
```

B.6 Datová zpráva

```
DataType ::= Byte(
   001 | //uint8
   002 | //uint16
   003 | //uint32
   004 | //uint64
   005 | //int8
   006 | //int16
   007 | //int32
   008 | //int64
   009 | //float
   010 | //double
   011 | //string
   012  //boolean
)


Data ::= SEQUENCE {DataType type}{
   dataType     type,
   data         SEQUENCE(SIZE(1..8)) of Byte,
   len          Byte(0..10),
   description  IA5String(Size(0..10)
}

ModuleData ::= SEQUENCE{
   moduleID Byte(0..255),
   numberOfData  Byte(0..8),
   data    SEQUENCE(Size(0..8)
}

DataMessage ::= SEQUENCE {
```

```
   header     MessageHeader{{004}},
   modules    SEQUENCE(SIZE(1..64)) of ModuleData
}
```

B.7 Datová zpráva se všemi daty z bufferu

```
DataType ::= Byte(
   001 | //uint8
   002 | //uint16
   003 | //uint32
   004 | //uint64
   005 | //int8
   006 | //int16
   007 | //int32
   008 | //int64
   009 | //float
   010 | //double
   011 | //string
   012  //boolean
)


Data ::= SEQUENCE {DataType type}{
   dataType     type,
   data         SEQUENCE(SIZE(1..8)) of Byte,
   len          Byte(0..10),
   description  IA5String(Size(0..10)
}

BufferData ::= SEQUENCE{
   numberOfData  Byte(0..8),
   data          SEQUENCE(Size(0..8) of Data
}


BufferModuleData ::= SEQUENCE{
   moduleID Byte(0..255),
   data     SEQUENCE(SIZE(1..5) of BufferData
}
```

```
BufferDataMessage ::= SEQUENCE{
   header     MessageHeader{{009}},
   modules    SEQUENCE(SIZE(1..64)) of ModuleData
}
```

B.8 Ukončovací zpráva

```
StopMessage ::= SEQUENCE {
   header     MessageHeader{{006}}
}
```

B.9 Zpráva kladného potvrzení

```
ACKMessage ::= SEQUENCE{
   header     MessageHeader{{007}}
}
```

B.10 Zpráva záporného potvrzení

```
NACKMessage ::= SEQUENCE{
   header     MessageHeader{{008}}
}
```

Příloha C: Protokol pro přenos dat do SmartCGMS

C.1 Hlavička protokolu

```
Opcode ::= uint8(
   001 | //Data
)


MessageHeader ::= SEQUENCE {Opcode: opcode}{
   magic       0xB33FB047
   opcode      opcode
   mesLen      uint16
}
```

C.2 Datová zpráva

```
DataIdentifier ::= uint8(
   000 | //Node
   001 | //Temperature_Skin
   002 | //Temperature_Ambient
   003 | //Position_Lat
   004 | //Position_Lon
   005   //Movement
)

Data ::= SEQUENCE {DataIdentifier: identifier}{
```

```
    dataIdentifier  identifier
    unixTimestamp   uint32
    value           double
}

DataMessage ::= SEQUENCE {
    header     MessageHeader{{001}}
    data       SEQUENCE(SIZE(1..5041)) OF DATA
}
```