



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

DEPARTMENT OF COMPUTER SYSTEMS

**HLASOVACÍ ZAŘÍZENÍ NA BÁZI BLUETOOTH LOW  
ENERGY**

BLUETOOTH LOW ENERGY VOTING DEVICE

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**TOMÁŠ VYBÍRAL**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. ZDENĚK VAŠÍČEK, Ph.D.**

BRNO 2017

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačových systémů

Akademický rok 2016/2017

**Zadání bakalářské práce**

Řešitel: **Vybíral Tomáš**

Obor: Informační technologie

Téma: **Hlasovací zařízení na bázi Bluetooth Low Energy  
Bluetooth Low Energy Voting Device**

Kategorie: Vestavěné systémy

Pokyny:

1. Seznamte se s technologií Bluetooth Low Energy (BLE).
2. Zpracujte studii na výše uvedené téma.
3. Formou prototypu navrhnete jednoduché hlasovací zařízení osazené minimálně jedním tlačítkem a signalizací pomocí LED diod nebo buzzeru komunikující pomocí BLE. Zařízení optimalizujete zejména na cenu.
4. Navrhnete aplikaci pro Windows, která bude umožňovat registraci hlasujících zařízení a následné hlasování. Aplikace musí umožňovat definovat poměr hlasu pro každého hlasujícího.
5. Zadání bude upřesněno po konzultaci s řešitelem.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů 1 a 2 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Vašíček Zdeněk, Ing., Ph.D.**, UPSY FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačových systémů a sítí  
602 00 Brno, Božetěchova 2



prof. Ing. Lukáš Sekanina, Ph.D.  
vedoucí ústavu

## Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací hardwarového zařízení a uživatelské aplikace pro hlasování. Pro komunikaci je využíváno protokolu Bluetooth Low Energy, kvůli jeho nízkoenergetickým vlastnostem. Výsledkem této práce je plně funkční prototyp periferie odesílající hlas uživatele do klientské aplikace, vytvořené pomocí frameworku Qt, která zpracovává jednotlivé hlasy a zobrazuje výsledky hlasování.

## Abstract

This bachelor's thesis deals with the design and implementation of embedded voting device and application for voting. The proposed voting device uses Bluetooth Low-Energy protocol which helps to achieve low energy consumption. The result of this thesis is a fully functional prototype. UI application was created using the Qt framework and is responsible for handling wireless connections, processing of individual voices and displaying the voting results.

## Klíčová slova

Bluetooth, Bluetooth Low Energy, BLE, UI, Qt, multiplatformní aplikace

## Keywords

Bluetooth, Bluetooth Low Energy, BLE, UI, Qt, cross-platform application

## Citace

VYBÍRAL, Tomáš. *Hlasovací zařízení na bázi Bluetooth Low Energy*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Vašíček Zdeněk.

# Hlasovací zařízení na bázi Bluetooth Low Energy

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Zdeňka Vašíčka, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Tomáš Vybíral  
10. května 2017

## Poděkování

Děkuji panu Ing. Zdeňku Vašíčkovi, Ph.D. za odborné vedení práce, cenné rady, vstřícnost při konzultacích a vypracovávání bakalářské práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>4</b>
<b>2</b>	<b>Bluetooth technologie</b>	<b>5</b>
2.1	Vývoj verzí	5
2.2	Technická specifikace	6
2.2.1	Frekvenční pásmo	6
2.2.2	Typy výkonnostních tříd	7
2.2.3	Architektura Bluetooth	7
2.3	Vytvoření spojení a párování	9
2.3.1	Topologie Bluetooth	9
2.3.2	Bluetooth adresa a jméno	9
2.3.3	Proces spojení	10
2.3.4	Párování a vázání (bonding)	10
<b>3</b>	<b>Bluetooth low energy</b>	<b>12</b>
3.1	Rozdíl oproti klasické technologii Bluetooth	12
3.2	Bluetooth low energy protocol stack	13
3.2.1	GAP – Generic Access Profile	13
3.2.2	GATT – Generic Attribute Profile	16
<b>4</b>	<b>Hardwarová podpora BLE</b>	<b>18</b>
4.1	Vybrané Cypress moduly	18
4.1.1	Popis použitých modulů	19
<b>5</b>	<b>Návrh a implementace</b>	<b>20</b>
5.1	Návrh hardware	20
5.2	Implementace programu pro BLE modul	21
5.3	Návrh a implementace uživatelského rozhraní	22
5.3.1	Hlavní okno aplikace	23
5.3.2	Připojení hlasovacího zařízení	27
5.4	Problémy při vývoji	28
<b>6</b>	<b>Ověření řešení</b>	<b>30</b>
6.1	Ověření funkčnosti hlasování	30
6.2	Ověření maximálního dosahu zařízení	31
6.3	Spotřeba periferního zařízení	32
6.4	Cena	33

<b>7 Závěr</b>	<b>35</b>
<b>Literatura</b>	<b>37</b>
<b>Přílohy</b>	<b>39</b>
Seznam příloh . . . . .	40
<b>A Schéma zapojení PProC modulu</b>	<b>41</b>
<b>B Obsah CD</b>	<b>42</b>

# Seznam obrázků

2.1	Bluetooth protocol stack architektura [10]	8
2.2	Piconet a scatternet (M – master, S – slave)	9
2.3	Rozdělení Bluetooth adresy	10
3.1	Advertising a skener [7]	14
3.2	Aktivní/Pasivní skenování [7]	15
3.3	BLE – Broadcastová topologie	15
3.4	BLE – Klient-server topologie	16
3.5	GATT – Hierarchie dat	17
4.1	Desky plošného spoje s BLE moduly [3, 4]	19
5.1	DPS pro vybraný PSoC modul	21
5.2	Konfigurace BLE zařízení	22
5.3	Hlavní okno aplikace	24
5.4	Vzhled stránky pro přidání uživatele a otázky	25
5.5	Náhled stránky pro hlasování	26
5.6	Dialogové okno pro připojení zařízení	28
6.1	Konfigurace BLE zařízení	31
6.2	Vysílání advertising paketu	32
6.3	Odesílání dat	33

# Kapitola 1

## Úvod

V současné době se bezdrátová komunikace neustále vyvíjí a rozšiřuje se do více odvětví. Hlavním důvodem je, že se bezdrátová komunikace stala spolehlivější, rychlejší a zvýšila se také bezpečnost přenášených dat. Využívá se jak v průmyslovém odvětví, tak i v běžném životě, například v současnosti pro velmi populární fitness senzory, které odesílají data do mobilního telefonu, kde si uživatel může zobrazit získané informace. Charakteristickým rysem těchto senzorů je malý datový přenos a hlavně nízká spotřeba energie, tak aby byla zaručena co nejdelší výdrž baterie.

Tato bakalářská práce je zaměřena na návrh a implementaci hlasovacího zařízení využívajícího technologii Bluetooth Low Energy, od společnosti Cypress, pro zaznamenávání jednotlivých hlasů uživatelů. Primárně je aplikace určena pro hlasování valné hromady akcionářů, kde dům má více než jednoho vlastníka a každý z nich různý vlastnický podíl. Aplikovat lze také pro obecná hlasování, kde je potřeba sčítat jednotlivé hlasy bez podílu. Cílem je vytvořit desktopovou aplikaci, která umožní přidávat vlastníky a přiřadit jim jejich vlastnický podíl, poté po spuštění hlasování zaznamenávat jejich hlasy, které jsou podle jejich váhy hlasu přepočítávány na procenta. Součástí je také navržení a vytvoření desky plošného spoje, která bude osazena třemi tlačítky určenými pro hlasování (ano, ne, zdržet se hlasování), Bluetooth modulem, umožňujícím tyto data odeslat do počítače pro zpracování a led diodou s malým reproduktorem umožňujícím poskytnout zpětnou vazbu uživateli.

Následující kapitoly se zabývají technologiemi Bluetooth a Bluetooth Low Energy, jejich funkcí a hlavními rozdíly mezi nimi. V další části je popsán návrh a implementace jak hardwarového zařízení, tak i klientské aplikace pro hlasování, která je naprogramována v multiplatformním frameworku Qt.



## Kapitola 2

# Bluetooth technologie

Tato kapitola je zaměřena na historii, vývoj a popis technologie Bluetooth. Je zde také popsána topologie sítě, která je vytvářena propojením několika Bluetooth zařízení a jejich proces spojení a párování.

Bluetooth je standard pro bezdrátovou komunikaci na krátké vzdálenosti s nízkou spotřebou energie, který umožňuje výměnu dat mezi dvěma nebo více elektronickými zařízeními. Vytvořen byl v roce 1994 firmou Ericsson, jako bezdrátová náhrada za sériové rozhraní RS-232. Vytváří jednoduché a stále spojení mezi velkou škálou komunikačních zařízení. Bluetooth funguje v nelicencovaném frekvenčním pásmu od 2.4 do 2.485 GHz. V současnosti najdeme Bluetooth ve většině přenosných zařízení jako jsou například mobilní telefony, tiskárny, bezdrátové klávesnice a myši. Využití lze nalézt také v průmyslovém odvětví, kde se používá pro přenos dat ze senzorů a v měřicích systémech.

### 2.1 Vývoj verzí

Jako každá technologie v dnešní době i Bluetooth se neustále vyvíjí. V prvních letech byly odstraňovány hlavně závažné chyby a doplňovány různé standardy. Všechny verze specifikace jsou zpětně kompatibilní [2, 19].

- **1.0** — V roce 1999 byla vydána specifikace Bluetooth 1.0. V prosinci téhož roku byla vydána verze 1.0b. Tyto verze měly mnoho problémů a výrobci měli potíže, aby jejich výrobky byly schopné spolupracovat. Objevovaly se především problémy s jednoznačným přiřazením rolí Master/Slave (viz kapitola 2.3.1) a kompatibilitou zařízení různých výrobců.
- **1.1** — Tato verze byla představena v únoru roku 2001, představovala základ pro první komerčně prodávané produkty. Byly odstraněny chyby z předchozích verzí. Byla také přidána podpora pro nechráněnou komunikaci.
- **1.2** — V této verzi byla technologie Bluetooth kompletně přepracována. Stále byla zachována zpětná kompatibilita, ale verze 1.2 poskytovala rychlejší připojení a vyhledávání zařízení. Vylepšena byla také odolnost vůči rušení, dále přidána rozsáhlejší podpora HID<sup>1</sup> zařízení, například myš a klávesnice.

---

<sup>1</sup>Human Interface Device Profile

- **2.0 + EDR<sup>2</sup>** — Tento standard pochází z roku 2004. Největší změnou bylo rozšíření EDR, pro rychlejší přenos dat. Udávaná rychlost byla až 3 Mbit/s, ale reálná přenosová rychlost se pohybovala okolo 2.1 Mbit/s.
- **2.1 + EDR** — Hlavní rysem je zavedení vlastnosti Secure Simple Pairing (SSP), která zvyšovala rychlost párování zařízení a bezpečnost tohoto procesu. Dalším vylepšením bylo Extended Inquiry Response (EIR), které poskytovalo více informací v průběhu vyhledávání, které umožnilo lepší třídění zařízení před připojením, což snížilo spotřebu energie.
- **3.0 + High speed** — V dubnu roku 2009 byla představena verze 3.0, podporující teoretickou rychlost přenosu dat až 24 Mbit/s. Bluetooth je zde využito pouze k navázání spojení a vysokorychlostní přenos se provádí přes souběžné spojení 802.11 známé jako Wi-Fi.
- **4.0** — Rok po představení předchozí verze se objevila specifikace Bluetooth 4.0. Cílem nebylo nahradit dřívější verzi 3.0, ale změnit způsob řízení spotřeby energie. Předpoklad byl, že se bude využívat pro malá zařízení, kde dochází jen k malému přenosu dat, což výrazně prodlouží životnost baterie. Pro přenos velkých objemů dat se měla využít specifikace verze 3.0. V této verzi se poprvé objevuje název Bluetooth Low Energy (BLE).
- **4.1** — Jedná se o softwarový update oproti verzi 4.0, který zvyšuje použitelnost zařízení. Update obsahoval podporu pro LTE<sup>3</sup> a bulk data exchange rates.
- **4.2** — Vyšla v roce 2014, zavádí funkcionalitu pro Internet of Things (IoT). Na rozdíl od specifikace 4.1, která byla aktualizována pouze softwarově, není zajištěna zpětná kompatibilita se všemi staršími čipy [8].
- **5.0** — Nejnovější specifikace, představena byla v červnu 2016. Nové vlastnosti jsou zaměřeny hlavně na IoT. Má čtyřnásobný dosah, 2-krát vyšší rychlost a poskytuje 8-krát větší kapacitu datové přenosnosti oproti Bluetooth 4.x, což může být důležité pro další rozvoj IoT.

## 2.2 Technická specifikace

### 2.2.1 Frekvenční pásmo

Bluetooth technologie využívá pro přenos dat rádiové vlny bezlicenčního ISM<sup>4</sup> pásma, které umožňuje použití této technologie po celém světě. Jedná se o pásmo 2.4 GHz, ve kterém je definováno 79 rádiových kanálů se šířkou pásma 1 MHz [13]. Šířka pásma se v některých zemích liší, odlišnosti v jednotlivých zemích zobrazuje tabulka 2.1.

---

<sup>2</sup>Enhanced data rate

<sup>3</sup>Long Term Evolution

<sup>4</sup>Industrial, Scientific, Medical

Země	Rozsah frekvence	RF kanály	
Evropa, USA a zbytek světa	2400 – 2483.5 MHz	$f = 2.402 + n$ (MHz)	$n = 0, \dots, 78$
Japonsko	2471 – 2497 MHz	$f = 2.473 + n$ (MHz)	$n = 0, \dots, 22$
Španělsko	2445 – 2475 MHz	$f = 2.449 + n$ (MHz)	$n = 0, \dots, 22$
Francie	2446.5 – 2483.5 MHz	$f = 2.454 + n$ (MHz)	$n = 0, \dots, 22$

Tabulka 2.1: Šířka pásma pro jednotlivé země [13]

## 2.2.2 Typy výkonostních tříd

Podle výstupního výkonu se Bluetooth zařízení dělí do jednotlivých tříd, viz tabulka 2.2. Vzdálenosti udávané v tabulce platí pouze ve volném prostoru, protože pokud jsou mezi komunikujícími zařízeními překážky (zdi, člověk) dosah rychle klesá. Dosah také závisí na kvalitě zařízení, konfiguraci antény a stavu baterie.

Třída	Max. povolený výkon		Dosah
	[mW]	[dBm]	
<b>1</b>	100	20	~100 m
<b>2</b>	2.5	4	~10 m
<b>3</b>	1	0	~1 m

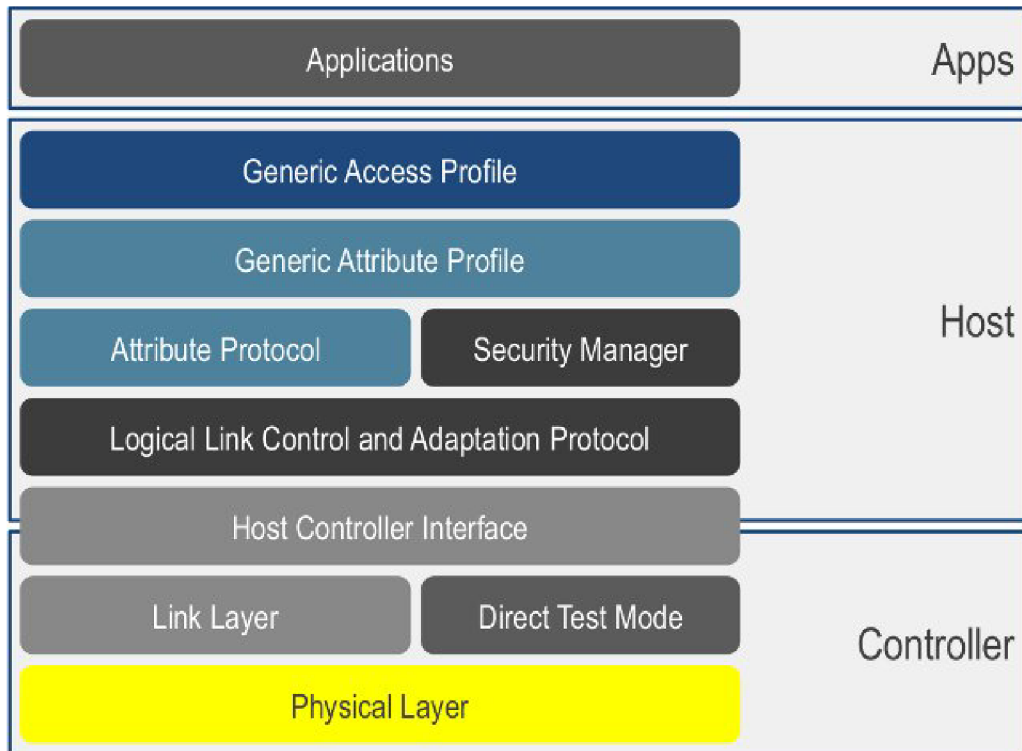
Tabulka 2.2: Rozdělení Bluetooth zařízení podle výkonosti [19]

## 2.2.3 Architektura Bluetooth

Technologie Bluetooth je definována na několika vrstvách od nejnižší rádiové vrstvy až po aplikační vrstvu (viz obrázek 2.1). Jednotlivé vrstvy jsou popsány v dalších kapitolách. Bluetooth stack můžeme rozdělit na dvě hlavní části [18, 16]:

- Bluetooth host stack** — Protokoly v této části pracují s daty na vyšší úrovni jako protokoly v controller stacku. Tyto protokoly jsou většinou implementované jako část operačního systému. Integrované zařízení, jako například handsfree, mohou mít oba tyto zásobníky implementované na jednom mikroprocesoru. Tato implementace vytváří tzv. *hostless systém*.
- Bluetooth controller stack** — Jedná se hardwarový modul, který řídí a určuje způsob komunikace. Skládá se ze tří základních částí:
  - rádiový vysílač (Bluetooth radio), který zajišťuje samotný rádiový přenos
  - linkový ovladač (Link Controller), který řídí navázání spojení, identifikaci a samotnou komunikaci mezi zařízeními
  - správce linky (Link Manager) a vstupně/výstupní obvody spoje, které zajišťují komunikaci s konkrétním zařízením

Bluetooth specifikace umožňuje spolupráci zařízení s různými systémy, definováním protokolových zpráv. Umožňuje také spolupráci mezi nezávislými podsystémy (Bluetooth controller, Bluetooth host) pomocí rozhraní. Toto rozhraní se nazývá *Host Controller Interface (HCI)*, přes které tyto podsystémy obousměrně komunikují. Dále v této kapitole jsou popsány jednotlivé vrstvy Bluetooth protocol stacku.



Obrázek 2.1: Bluetooth protocol stack architektura [10]

- **Physical Layer** — Kontroluje přenos a přijímání radiových vln za pomoci komunikačních kanálů.
- **Link Layer** — Sestavuje, kontroluje a ukončuje spojení. Definiuje strukturu a kanály paketů a zajišťuje bezpečnost (správu klíčů, párování zařízení a řízení šifrování).
- **Direct Test Mode** — Umožňuje testování fyzické vrstvy, udělením příkazu pro odeslání nebo přijetí sekvence paketů. Pro odeslání příkazů je využita vrstva HCI.
- **Host Controller interface** — Standardizované komunikační rozhraní spojující dvě hlavní části Bluetooth controller stack a Bluetooth host stack. Zajišťuje jednotný přístup k funkcím Link Manageru. Využívá různá hardwarová rozhraní k transformaci stejných příkazů a paketů.
- **Logical Link and Adaptation Protocol** — Též známý jako *L2CAP* posílá pakety na rozhraní HCI a nebo přímo Link Manageru (hostless systémy). Dále zajišťuje rozdělení a znovu složení dlouhých paketů.
- **Attribute Protocol** — Při úspěšném spojení zařízení definuje klient-server protokol pro výměnu dat. Atributy jsou seskupeny do služeb s využitím Generic Attribute protokol. Využíván je hlavně v LE<sup>5</sup> implementacích.
- **Security Manager** — Definiuje protokol pro párování, ověřování a šifrování dat mezi zařízeními. Poskytuje bezpečnostní funkce, které využívají ostatní části.

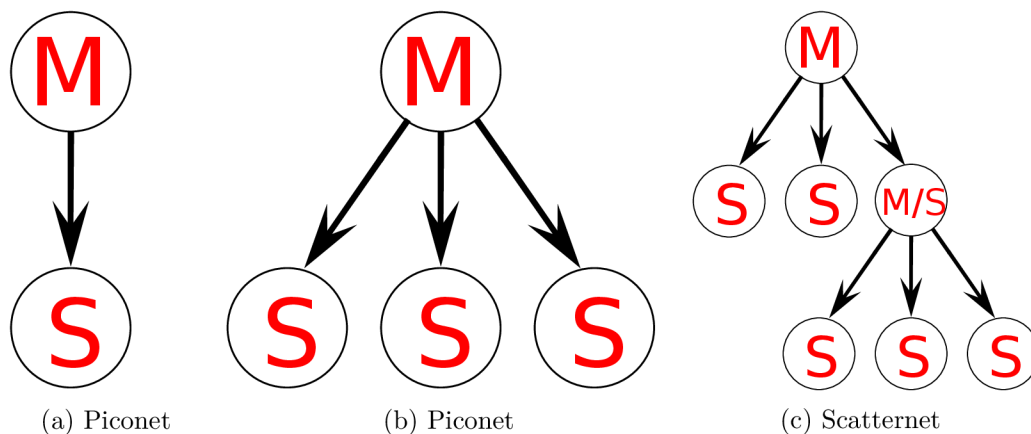
<sup>5</sup>Low Energy

Vrstvy Generic Access Profile a Generic Attribute Profile jsou využívány v Bluetooth Low Energy. Jejich podrobný popis je v kapitolách 3.2.1 a 3.2.2.

## 2.3 Vytvoření spojení a párování

### 2.3.1 Topologie Bluetooth

Bluetooth využívá hvězdicovou topologii sítě. Dvě zařízení sdílející stejný kanál tvoří jednotku zvanou *piconet* (viz obrázek 2.2a). Piconet se skládá maximálně z osmi zařízení, jednoho centrálního a sedmi periferních zařízení. Na centrální zařízení *master* se připojují zařízení typu *slave*. Přičemž master může komunikovat až se sedmi zařízeními najednou, ale slave může komunikovat pouze s jedním zařízením typu master (viz obrázek 2.2b), podrobnější vysvětlení v kapitole 3.2.2. Protokol umožňuje výměnu rolí master/slave (v momentě, kdy nekomunikují). Tyto vlastnosti pak umožňují topologické uspořádání nazvané *scatternet* (viz obrázek 2.2c). Každý piconet komunikuje na jiném kanálu s jinou sekvencí změn frekvence [14].

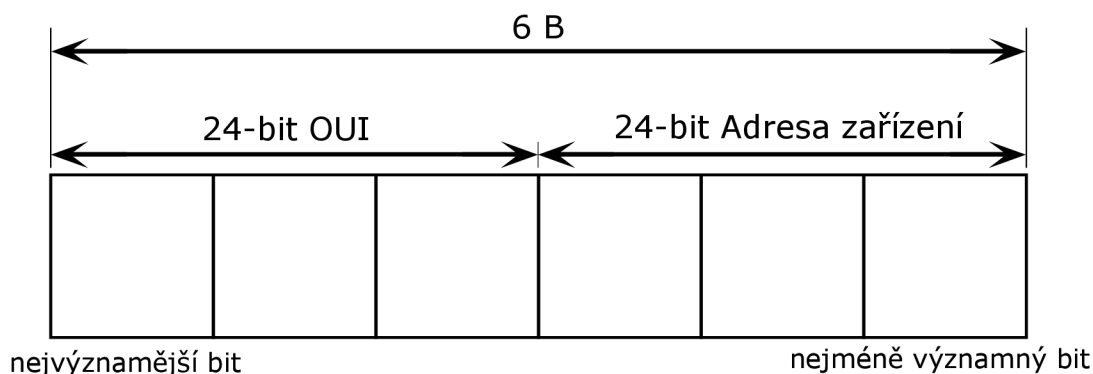


Obrázek 2.2: Piconet a scatternet (M – master, S – slave)

### 2.3.2 Bluetooth adresa a jméno

Všechny Bluetooth zařízení mají jedinečnou 48 bitovou adresu. Obvykle je zkrácena jako *BD\_ADDR*. Délka adresy je většinou 12 hexadecimálních číslic. Horní polovina (24 významnějších bitů) je jedinečná adresa organizace (OUI<sup>6</sup>). Zbýlých 24 bitů je konkrétní adresa zařízení, viz obrázek 2.3.

<sup>6</sup>Organization Unique Identifier



Obrázek 2.3: Rozdělení Bluetooth adresy

Aby uživatel dokázal lépe rozpoznat jednotlivá zařízení, lze jim také přiřadit jméno, které je zobrazováno ve většině případů místo hexadecimální adresy. Pravidla jsou zde již méně striktní. Maximální délka je 248 bytů a dvě zařízení mohou mít stejné jméno.

### 2.3.3 Proces spojení

Vytváření spojení mezi dvěma zařízeními je víceetapový proces, obsahující tři stavy:

1. **Dotazování (Inquiry)** — Pokud obě zařízení neví nic o ostatních zařízeních, jedno ze zařízení pošle Inquiry požadavek (snaží se najít další zařízení). Když druhé zařízení tento požadavek zaregistruje, odešle zpět zprávu se svojí adresou, případně se svým jménem a ostatními informacemi.
2. **Připojování (Connecting)** — Zde se vytváří spojení mezi dvěma Bluetooth zařízeními. Je nutné, aby zde obě zařízení znala adresu druhého zařízení, ke kterému se má připojit. Zjištění adresy je zajištěno předchozím procesem.
3. **Spojení (Connection)** — Po dokončení procesu připojování, se zařízení nachází v jednom ze čtyř módů:
  - **Active Mode** — V tomto módu si zařízení aktivně vyměňují data.
  - **Sniff Mode** — Mód je určen pro šetření energie, když je zařízení méně aktivní je usnává (například na 100 ms) a čeká na přenos dat.
  - **Hold Mode** — Hold mód je dočasný šetřič energie, kdy zařízení spí po zadaný časový interval. Zpět do aktivního módu se vrací po uplynutí intervalu.
  - **Park Mode** — Zde je zařízení v hlubokém spánku. Master může přikázat slave zařízení, aby přešlo do tohoto módu. Slave zařízení je neaktivní dokud mu master nepřikáže, aby se opět vzbudilo.

### 2.3.4 Párování a vázání (bonding)

Dvě vázané zařízení se automaticky připojují pokud jsou dostatečně blízko. Není potřeba interakce s uživatelem. V dnešní době je využíváno například pro automatické propojení smartphonu a automobilu, smartphonu a bezdrátových sluchátek, atd.

Pro vytvoření vazby je nutné poprvé spustit proces párování. Spárovaná zařízení sdílejí mezi sebou adresy, jména a profily, které ukládají do paměti. Zařízení si také ukládají tajný klíč, který použijí, když se budou chtít automaticky spojit.

Párování obvykle vyžaduje ověření, kde uživatel musí potvrdit spojení mezi zařízeními. Postup při ověřování se může lišit a obvykle závisí na možnostech rozhraní jednoho nebo druhého zařízení. Někdy stačí pouze stisknout tlačítko, někdy je zapotřebí zadání PIN kódu na každém zařízení. Pin kód může mít různou složitost a délku, od čtyř číslic až po šestnáct znaků.

## Kapitola 3

# Bluetooth low energy

V této kapitole je popsán hlavní rozdíl oproti klasické technologii Bluetooth. Poté je zaměřena hlavně na popis dvou podkategorií Bluetooth protocol stacku, konkrétně na Generic Access Profile (GAP) a Generic Attribute Profile (GATT).

Bluetooth Low Energy (BLE), dříve známý jako Bluetooth Smart, byl vyvinut pro Internet of Things (IoT). Jde o univerzální bezdrátový standard pro jednoduchou výměnu dat mezi dvěma zařízeními. Zařízení využívající BLE může díky jeho vlastnostem být v provozu i několik let na mincové baterii. Podpora technologie Bluetooth na každém operačním systému umožňuje vývoj pro široký rozsah zařízení, od domácích spotřebičů, bezpečnostních systémů, senzorů po fitness monitory [1, 17].

### 3.1 Rozdíl oproti klasické technologii Bluetooth

Hlavní rozdíl vyplývá už z názvu. BLE se hlavně využívá v zařízeních, kde se přenášejí jen malé objemy dat, takže spotřeba energie je velmi nízká. V dnešní době se jedná hlavně o senzory, například pro měření tepu, kdy data jsou odesílána do mobilního telefonu pro zpracování. Dalším rozdílem je počet kanálů. Jak je výše uvedeno, klasická technologie Bluetooth má 79 kanálů. BLE má pouze 40 kanálů a poslední tři jsou tzv. *advertising kanály*, které jsou využity pro přenos informací o zařízení. Liší se také ve vytváření spojení, kde klasická technologie Bluetooth vytváří jednotky tzv. *piconety* (viz 2.2), maximální počet připojených zařízení je sedm, u BLE technologie tento počet není specifikován, závisí hlavně na verzi BLE stacku a hardwarové implementaci.

Bluetooth low energy, jak jsem již zmínil, je vyvíjen s důrazem na spotřebu, proto se některé parametry liší oproti klasické technologii Bluetooth. Přehled hlavních rozdílů je uveden v tabulce 3.1.



	Bluetooth Classic	Bluetooth Low Energy
<b>Frekvence</b>	2.4 GHz	2.4 GHz
<b>Maximální propustnost</b>	1–3 Mbit/s	1 Mbit/s
<b>Typická propustnost</b>	0.7–2.1 Mbit/s	0.27 Mbit/s
<b>Přibližný dosah</b>	10 m (class 2), 100 m (class 1)	50 m
<b>Výdrž baterie</b>	dny	měsíce–roky
<b>Max. počet slave zařízení</b>	7	nedefinováno

Tabulka 3.1: Porovnání klasické technologie Bluetooth a Bluetooth Low Energy [9]

## 3.2 Bluetooth low energy protocol stack

Protokol stack klasické technologie Bluetooth je popsán v kapitole 2.2.3. Jak jsem již zmiňoval, protocol stack je rozdělen do dvou hlavních kategorií. Každá z těchto kategorií má podkategorie, které provádějí specifické úlohy. Dvě podkategorie, které jsou dále popsány jsou Generic Access Profile (GAP) a Generic Attribute Profile (GATT) [12, 17].

### 3.2.1 GAP – Generic Access Profile

Generic Access Profile je základním kamenem umožňujícím BLE zařízením mezi sebou komunikovat. Aby zařízení od různých výrobců mohly spolupracovat, GAP specifikuje jak se mají BLE zařízení chovat při vyhledávání ostatních zařízení, odesílání dat, navazování bezpečného spojení, apod.

#### Role zařízení

- **Vysílání** — Není potřeba explicitně vytvářet spojení pro přenos dat.
  - **Broadcaster** — Zařízení odesílá veřejné advertising pakety.
  - **Observer** — Zařízení přijímá data, které odeslal Broadcaster.
- **Připojení** — Zde musí zařízení explicitně vytvářet spojení pro výměnu dat. Tyto role jsou více běžně využívány než vysílací role.
  - **Peripheral** — Peripheral role odpovídá *slave* zařízením. Používá advertising pakety, umožňující centrálním zařízením je nalézt a následně navázat spojení. Po úspěšném připojení, se pakety přestávají posílat a zařízení zůstává připojené k centrálnímu zařízení, které přijalo jeho žádost. Většinou se jedná o zařízení s nízkou spotřebou energie, které odesílá data nebo signál, že je stále aktivní.
  - **Central** — Tato role odpovídá *master* zařízením. Jsou iniciátorem a správci spojení. Přijímají advertising pakety vysílané peripheral zařízeními. V jeden okamžik můžou připojit několik peripheral zařízení. Udržení více spojení je více náročné na výpočetní výkon, obvykle se jedná o notebooky, smartphony nebo tablety.

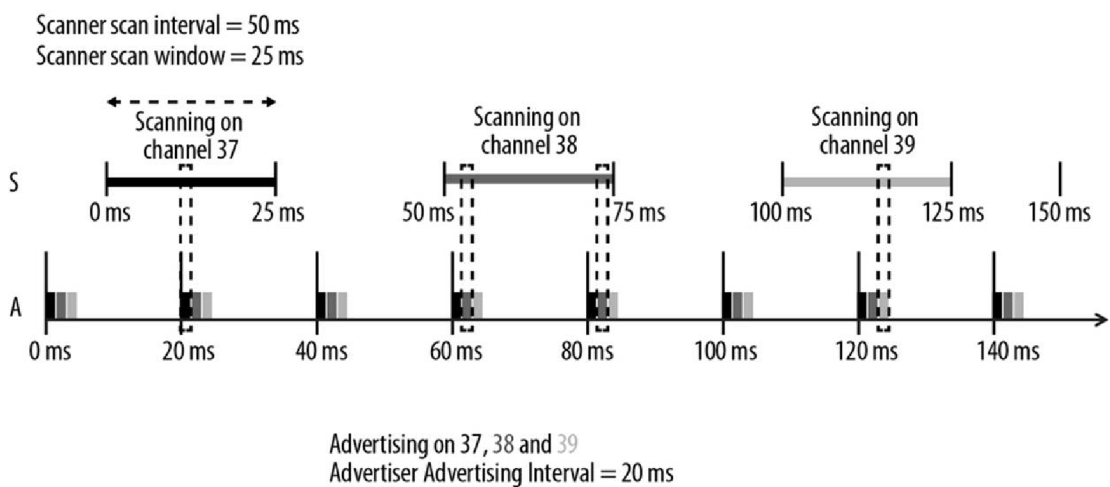
#### Advertising a skener

BLE má pouze jeden formát paketu a dva typy paketu (advertising a datové pakety). Advertising pakety slouží k těmto účelům:

1. Posílat data pro aplikace, které nepotřebují navazovat spojení (Broadcaster/Observer).
2. Nalezení peripheral (slave) zařízení a připojení k nim.

Každý advertising paket může obsahovat až 31 bytů advertising dat, společně se základní hlavičkou s informacemi, která obsahuje adresu zařízení. Pakety jsou odesílány v pravidelném *advertising intervalu*. Rozsah tohoto intervalu je od 20 ms do 10.24s. Čím kratší je interval, tím vyšší je šance na zachycení paketu centrálním zařízením, protože pakety jsou vysílány s vyšší frekvencí, což má za následek vyšší spotřebu energie.

Advertising používá maximálně tři frekvenční kanály. Advertiser a skener nejsou vzájemně synchronizováni, takže advertising paket je zachycen skenerem pouze v případě, že zrovna naslouchá (viz obrázek 3.1).

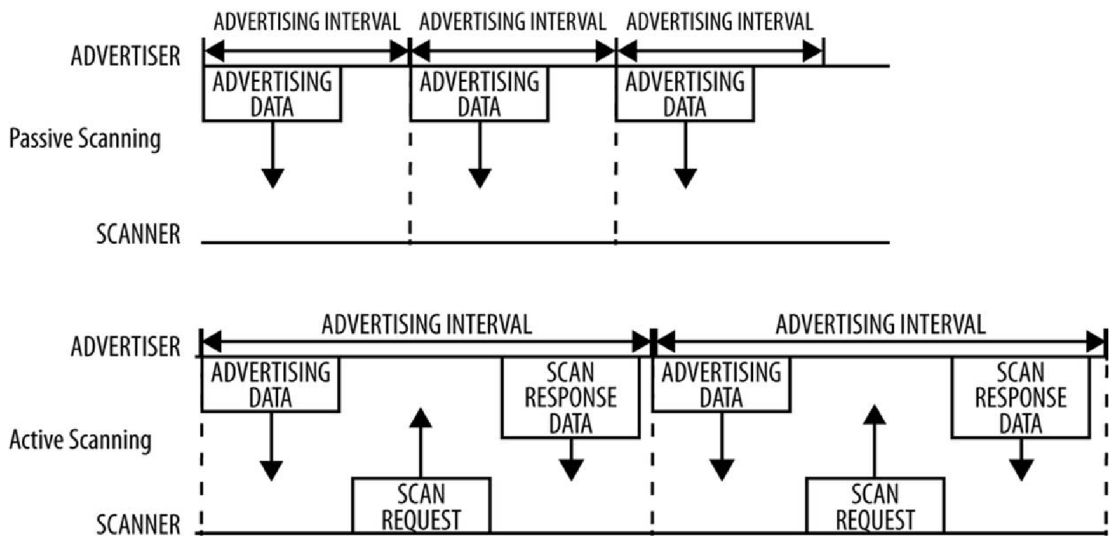


Obrázek 3.1: Advertising a skener [7]

Specifikace definuje dva základní typy skenování:

- **Pasivní** — Skener pouze přijímá pakety a advertiser neví, jestli byly pakety doručeny nebo se někde ztratily.
- **Aktivní** — V tomto typu režimu skener navíc odesílá *Scan Request* paket po obdržení advertising paketu. Advertiser na tento paket odpoví *Scan Response* paketem. Pomocí tohoto paketu je možné zdvojnásobit množství odesílaných dat.

Na obrázku 3.2 je znázorněn rozdíl mezi pasivním a aktivním skenováním.

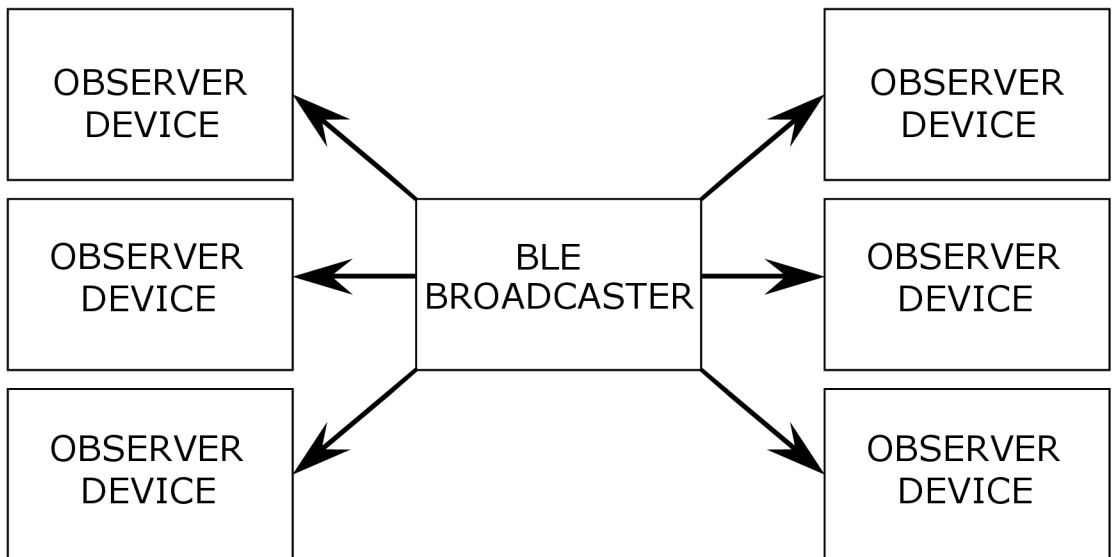


Obrázek 3.2: Aktivní/Pasivní skenování [7]

### Broadcasting a observing

Vytvořením broadcaster zařízení lze jednoduše vysílat data, tyto data může přijímat libovolný počet zařízení, protože zařízení mezi sebou nenavazují spojení. Broadcasting je v GAP definován jako všesměrové vysílání, kterého lze dosáhnout výhradně jen s pomocí *advertising dat*.

Observer aktivně vyhledává tyto pakety a přijímá je. V případě, že by se observer připojil na broadcastové zařízení, odesílání advertising paketu se zastaví a broadcaster přestane vysílat data. Tato situace nastane z důvodu výhradního spojení se slave zařízením.



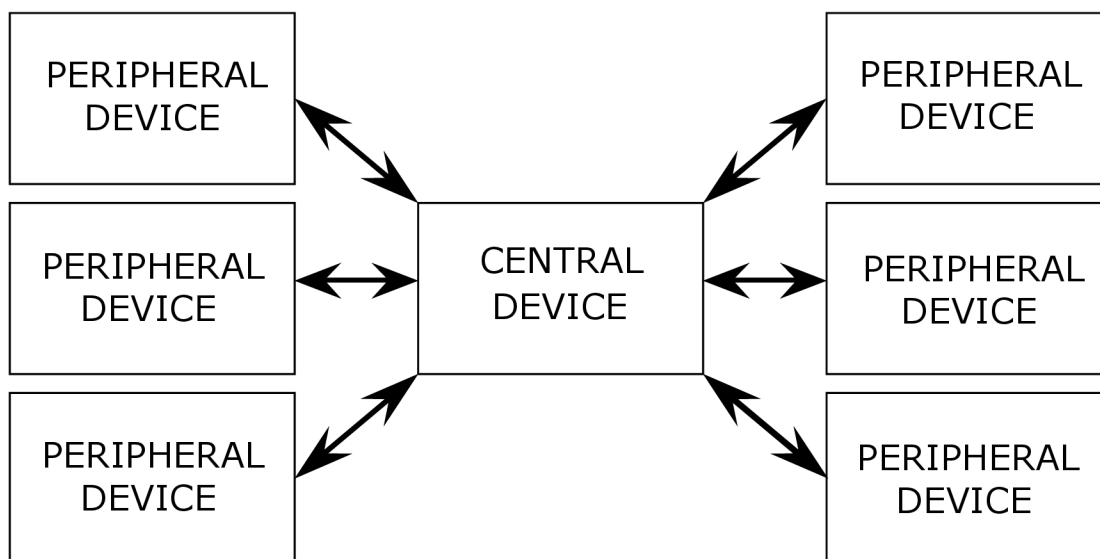
Obrázek 3.3: BLE – Broadcastová topologie

### 3.2.2 GATT – Generic Attribute Profile

GATT je zaměřen na reprezentaci dat, jak si budou dvě BLE zařízení vyměňovat data pomocí služeb (service) a charakteristik. Využívá k tomu *Attribute Protocol*, popisující strukturu uložených služeb a charakteristik. Všechny tyto informace ukládá do vyhledávací tabulky. GAP a GATT role jsou na sobě nezávislé, tedy peripheral nebo centrální zařízení může být klient nebo server, záleží na směru toku dat.

#### Role zařízení

- **Klient** — Centrální zařízení, které posílá požadavky na server a přijímá od něj odpovědi. Připojení na server je definované jako výhradní, proto server může být současně připojený jen k jednomu klientovi. Klient musí nejprve provést vyhledání služeb (service) poskytovaných serverem, aby mohl číst, zapisovat nebo přijímat aktualizace o změně hodnoty.
- **Server** — Periferní zařízení, které obsahuje Attribute Protocol, umožňující zpřístupnění dat klientovi. Přijímá požadavky od klienta a posílá zpět požadovaná data. Posílá klientovi echo o tom, že došlo ke změně dat na serveru. Každé BLE zařízení musí obsahovat alespoň základní server, a to i v případě, že bude vracet chybnou odpověď.



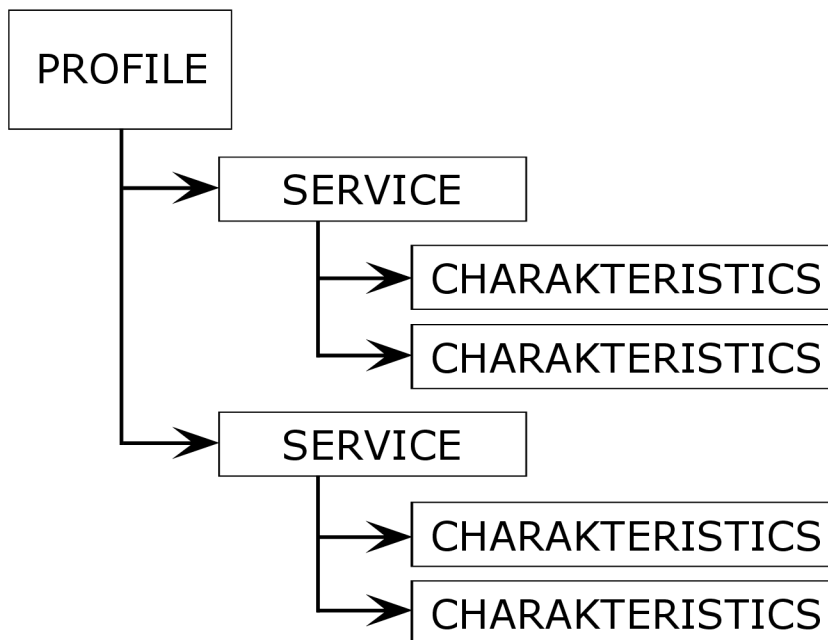
Obrázek 3.4: BLE – Klient-server topologie

#### Profily, služby a charakteristiky

BLE technologie má seznam předem definovaných služeb. Každá služba je schválena Bluetooth SIG<sup>1</sup>. Profil je rozdělen službami na logické části. Každá služba je předem popsána use case diagramem a rolemi. Toto umožňuje vyvíjet aplikaci využívající konkrétní službu a bude zajištěno, že aplikace bude fungovat s jakýmkoliv zařízením naprogramovaným touto službou. GATT umožňuje také vytvářet vlastní služby. Bluetooth používá pro označení služeb

<sup>1</sup>Special Interest Group

a charakteristik tzv. *UUID*<sup>2</sup> (viz dále), pomocí kterého je lze identifikovat. Služby obsahují charakteristiky, které představují nejnižší úroveň GATT transakcí. Každá charakteristika obsahuje dvě základní části, v první části jsou ukládána data a druhá, nepovinná část, slouží pro popis tzv. *Descriptor*.



Obrázek 3.5: GATT – Hierarchie dat

## UUID

UUID neboli univerzální unikátní identifikátor je 128 bitové číslo zajišťující globální unikátnost. Existuje také zkrácená verze z důvodu efektivnosti, která má pouze 32 bitů. Tento zkrácený formát může být použit pouze pro služby definované v Bluetooth specifikaci. UUID slouží při připojování periferního zařízení k nalezení a následnému připojení ke konkrétní službě a charakteristice, pomocí kterých jsou přenášena data.

---

<sup>2</sup>Universally unique identifier

## Kapitola 4

# Hardwarová podpora BLE

V této kapitole je popsán a zdůvodněn výběr vhodného BLE modulu pro implementaci hlasovacího zařízení a podrobně rozepsány vlastnosti vybraných modulů. Je zde také uveden popis multiplatformní aplikace Qt, použité pro vytvoření grafického uživatelského rozhraní.

Podporu pro BLE lze dnes nalézt v každém moderním operačním systému (Android, iOS, Windows, Linux) s verzí Bluetooth 4.0 a vyšší. Také výrobci vestavěných systémů nezaostávali a krátce po příchodu BLE se na trhu začaly objevovat první moduly podporující právě tuto technologii. V dnešní době existuje mnoho firem a nepřeborné množství nabízených modulů s různými vlastnostmi.

Při výběru BLE modulů jsem se zaměřil na známé celosvětové firmy, jako jsou Cypress, Texas Instruments a Nordic Semiconductur. Rozhodoval jsem se na základě nabízeného vývojového prostředí, které by mělo být jednoduché a co nejvíce usnadnit programování v něm, dále také podle podpory nabízené na internetu (FAQ, fórum, existující projekty). Hlavním faktorem byla cena a vlastnosti nabízených BLE modulů. Navrhované zařízení pracuje se třemi tlačítky, dvěma ledkami a jedním reproduktorem. Do centrálního zařízení se odesílá pouze informace o tom, které tlačítko bylo zmáčknuto, proto jsem se zaměřil na méně výkonná zařízení s nízkou spotřebou energie. Rozhodl jsem se pro firmu Cypress, která nabízí nejen uživatelsky přívětivé vývojové prostředí, ale také lze na internetu nalézt mnoho návodů jak začít a mnoho volně dostupných funkčních řešení využívajících právě BLE modul. Navíc veškerý software potřebný pro vývoj je dostupný zdarma, je vyžadována pouze registrace uživatele. Pro programování modulů je možno využít programovací jazyk „C“, který je doporučený nebo přímo assembler.

### 4.1 Vybrané Cypress moduly

Po dohodě s vedoucím bakalářské práce jsem se rozhodl pro dva různé moduly, které se liší hlavně v dosahu. Moduly od společnosti Cypress jsou rozděleny do dvou skupin:

- P<sub>Ro</sub>C (Programmable Radio on Chip)
- P<sub>So</sub>C (Programmable System on Chip)

Hlavní rozdíl mezi nimi je, že P<sub>So</sub>C čipy obsahují čtyři UDB (Universal Digital Block) jednotky, které může uživatel využít například jako časovač, čítač, pseudonáhodný generátor čísel, apod. Mají také integrovanou podporu pro řízení LCD displejů a mnoho dalších výhod. Proto se pro složitější řešení využívají právě moduly z této skupiny, které navíc vynikají flexibilitou.

Z jednotlivých skupin byly vybrány moduly s tímto označením:

- P<sub>RoC</sub> – CYBLE-222014-01 (viz obrázek 4.1a)
- P<sub>SoC</sub> – CYBLE-224110-00 (viz obrázek 4.1b)

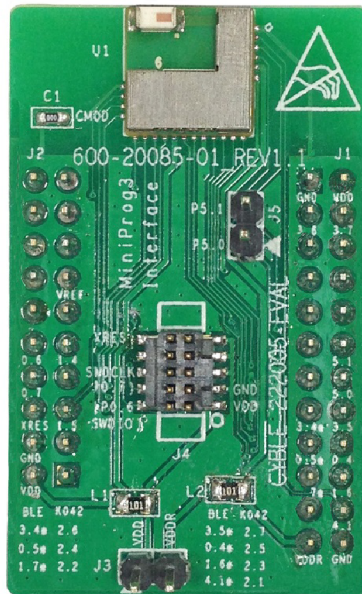
#### 4.1.1 Popis použitých modulů

Oba moduly obsahují často používaný a energeticky úsporný 32-bitový procesor ARM Cortex-M0 s maximální taktovací frekvencí 48 MHz. Flash paměť má velikost 256 KB a RAM paměť má velikost 32 KB. Další odlišnosti mezi moduly popisuje tabulka 4.1.

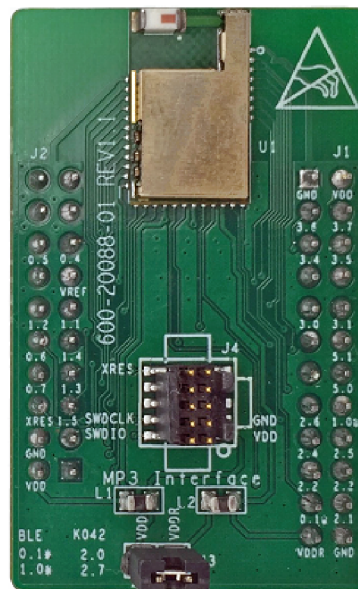
	P <sub>RoC</sub> modul	P <sub>SoC</sub> modul
<b>Verze Bluetooth</b>	4.2	4.1
<b>Udávaný dosah</b>	30 m	400 m
<b>Síla signálu</b>	+3 dBm	+9.5 dBm
<b>Spotřeba při vysílání</b>	15,6 mA	
<b>Spotřeba při přijímání</b>	16,4 mA	
<b>Spotřeba v stand-by režimu</b>	1,3 $\mu$ A	
<b>Napájecí napětí</b>	1.8 – 5.5 V	2.0 – 3.6 V
<b>GPIO</b>	16 pinů	25 pinů

Tabulka 4.1: Porovnání P<sub>RoC</sub> a P<sub>SoC</sub> modulu

V tabulce 4.1 jsou porovnávány pouze základní a informace důležité pro tuto práci. Moduly nabízejí mnoho dalších vlastností a funkcí, jako je například vestavěná podpora ovládání LCD, ale tyto možnosti modulů nejsou zde využity. Kompletní specifikaci modulů lze nalézt v datasheetu výrobce (viz [5, 6]).



(a) P<sub>RoC</sub> modul



(b) P<sub>SoC</sub> modul

Obrázek 4.1: Desky plošného spoje s BLE moduly [3, 4]

## Kapitola 5

# Návrh a implementace

V této kapitole je podrobněji popsán návrh hardwaru využitého pro hlasovací zařízení, implementace programu pro BLE modul, a také je zde popsáno jak probíhal návrh a implementace grafické uživatelské aplikace určené pro zaznamenávání jednotlivých hlasů.

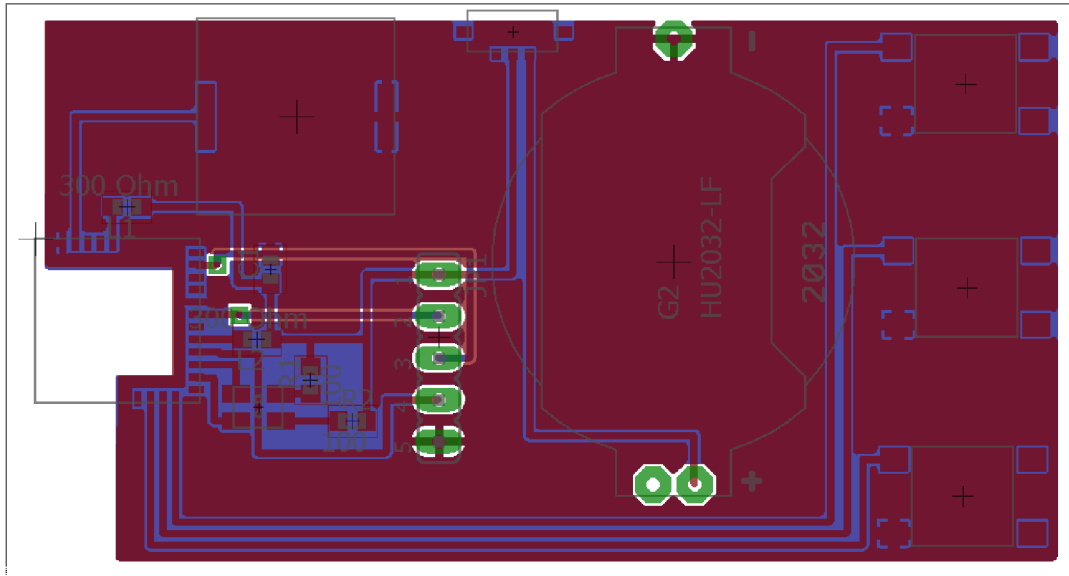
### 5.1 Návrh hardware

Pro návrh a následné vytvoření desky plošného spoje (DPS) jsem zvolil program Eagle, který je volně dostupný ke stažení. Pro tento program jsem se rozhodl z důvodu předchozích zkušeností, a protože výsledná DPS je velmi jednoduchá, bylo by zbytečné porovnávat vlastnosti a vymoženosti ostatních programů sloužících pro tento účel.

Ještě před začátkem návrhu schématu a DPS jsem si musel vytvořit schématickou značku pro BLE modul. V datasheetu výrobce lze nalézt přímo rozměry a umístění jednotlivých pinů pro modul. Nyní jsem měl připravenou schématickou značku pro modul, ale dále bylo potřeba upravit rozměry a umístění pinů pro tlačítka a spínač, protože mnou vybrané součástky neměly v Eaglu předem vytvořenou knihovnu. Ostatní součástky jsou implementovány přímo v Eaglu, nebo je lze nalézt volně dostupné na internetu.

Po tomto kroku již bylo možné vytvořit schéma a desku plošného spoje. Tlačítka jsou připojena k pinům za pomoci pull-up rezistoru. BLE modul tento odpor již obsahuje, proto není potřeba tento rezistor v schématu uvažovat. Pro připojení napájecího napětí je opět v datasheetu návod. Cypress zde přidává i tzv. *Ferrite bead* pro odstranění nežádoucího šumu. Poté již stačilo připojit k pinům BLE modulu přes předřadný odpor dvě led diody a reproduktor. Vytvořené schémata lze nalézt v příloze [A](#). Z vytvořeného schématu byla vytvořena DPS, kterou lze vidět na obrázku [5.1](#).



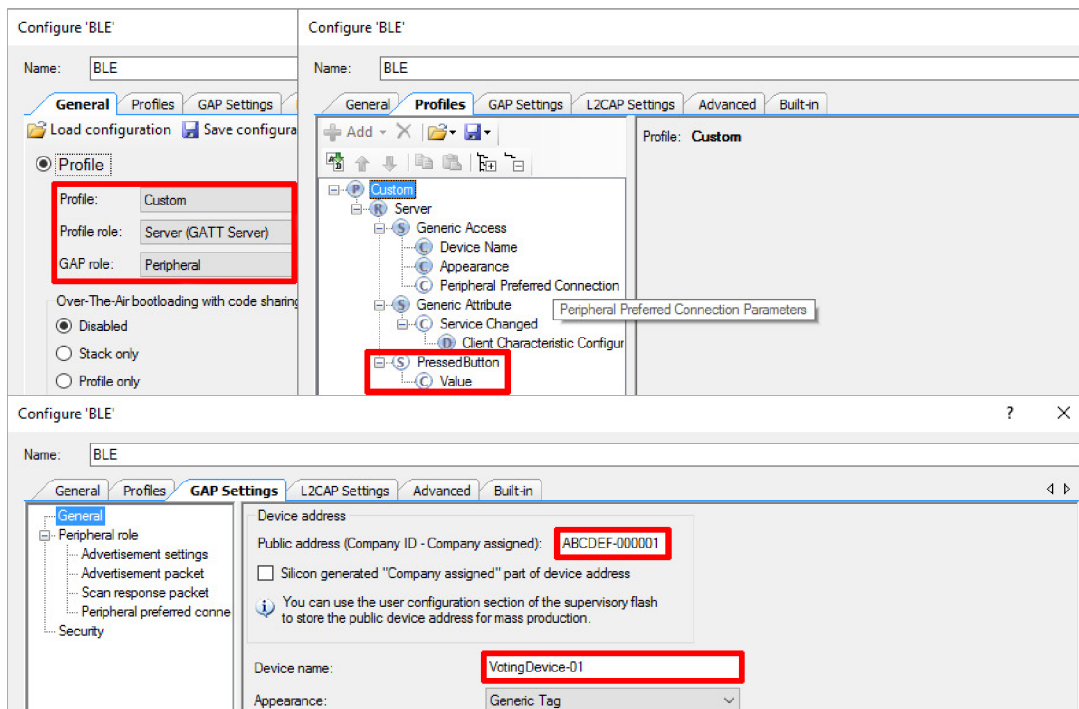


Obrázek 5.1: DPS pro vybraný PSoC modul

## 5.2 Implementace programu pro BLE modul

Jak jsem zmínil v předchozí kapitole, moduly jsou z různých skupin, ale pro programování se liší pouze v nastavení cílového modulu pro projekt a v jiném číslování pinů pro připojení periférií.

Jako první bylo potřeba správně nastavit konfiguraci BLE modulu, jestli se jedná o centrální nebo periferní zařízení, profil a GATT roli. Zařízení se bude připojovat pouze k jednomu centrálnímu zařízení, na které bude odesílat data, proto se jedná o periferní zařízení a GATT role je typu server (viz kapitola 3.2.2). Dalším krokem bylo vytvoření vlastního profilu, protože již existující profily obsahují mnoho služeb a charakteristik, které by zůstaly nevyužity. Vytvořil jsem tedy nový profil, který kromě základních služeb a charakteristik sloužících pro identifikaci a připojení zařízení, obsahuje jednu další službu s jednou charakteristikou, do které bude ukládána hodnota stisknutého tlačítka. Jednotlivá tlačítka pro hlasování jsou reprezentována hexadecimálními hodnotami ano (0x01), ne (0x02), zdržet se (0x03). Hodnota 0x00 je využita pro signalizaci, kdy nebylo stisknuto žádné tlačítko, nastavována je například při připojení zařízení (viz kapitola 5.3). Při nastavení hodnoty stisknutého tlačítka na hodnotu 0x11 je vlastníkovu povoleno hlasování, které je signalizováno rozsvícením zelené led diody, pokud není nastavena tato hodnota, aplikace nereaguje na stisk hlasovacího tlačítka. Posledním krokem konfigurace bylo nastavení MAC adresy a názvu zařízení. Všechna tato nastavení lze jednoduše provádět pomocí jednoduchého a přehledného dialogového okna (viz obrázek 5.2).



Obrázek 5.2: Konfigurace BLE zařízení

Hlavní program vždy na začátku provede inicializaci, a potom již běží v nekonečné smyčce, ve které se kontrolují stisknutá tlačítka a provádí se příslušné akce. Při inicializaci je registrována funkce, která je volána vždy, pokud dojde k události na BLE stacku (inicializace BLE stacku, připojení/odpojení zařízení, spuštění/ukončení odesílání advertising paketu). Událost připojení nebo odpojení zařízení je signalizována pomocí dvou led diod. Při úspěšném spojení bliká zelená led dioda a při odpojení zařízení červená. Slouží také pro signalizaci, že je povoleno hlasování a to pomocí rozsvícení zelené led diody. Při úspěšné inicializaci BLE stacku nebo při odpojení zařízení je spuštěno vysílání advertising paketu (viz kapitola 3.2.1).

Hlavní smyčka reaguje na stisk jakéhokoliv tlačítka pro hlasování. Při stisknutí jednoho ze tří tlačítek, je na 200 ms spuštěn reproduktor, což ve výsledku zazní jako jednoduché pípnutí, které slouží uživateli jako zpětná vazba, že bylo stisknuto tlačítko, navíc je hodnota stisknutého tlačítka uložena do charakteristiky, která je odeslána do centrálního zařízení, kde je tato hodnota dále zpracovávána. Pro úsporu energie je hodnota odesílána pouze v případě, že stisknuté tlačítko je odlišné od toho předchozího, a také hlavní cyklus se opakuje v intervalu 50 ms. Předpokládá se, že uživatel nedokáže stisknout tlačítko po dobu kratší než je tato hodnota a tím je zajištěno, že stisknuté tlačítko bude vždy zaznamenáno.

### 5.3 Návrh a implementace uživatelského rozhraní

Nejdříve bylo potřeba zjistit, co všechno aplikace musí umět a jaké musí mít vlastnosti, aby byla použitelná pro daný účel. Při návrhu aplikace jsem se nechal inspirovat již existující aplikací sloužící pro tento účel viz [11]. Odzkoušením tohoto programu jsem zjistil většinu věcí, které je potřeba implementovat, na co se zaměřit, a co přesně vyhledávat na internetu.

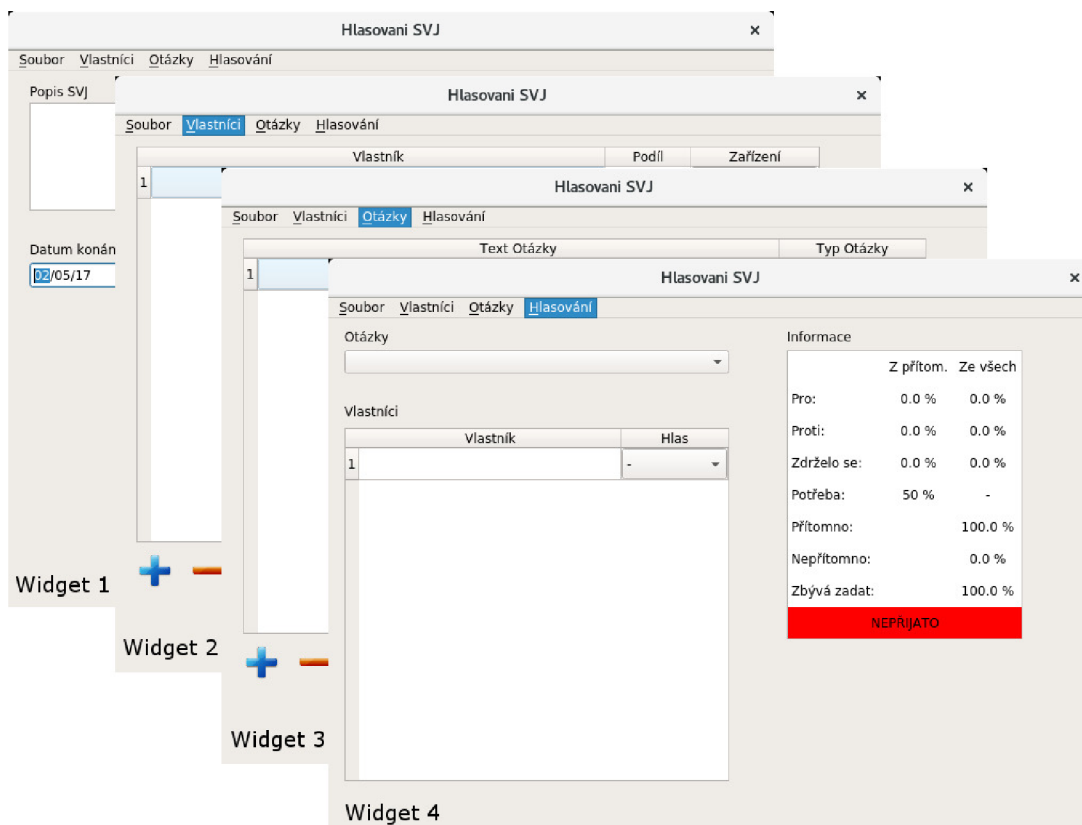
Požadované vlastnosti programu:

- zadání základních informací o hlasování
- správa vlastníků a nastavení jejich vlastnického podílu
- správa hlasovací zařízení pro jednotlivé vlastníky
- správa otázek, které budou pokládány účastníkům
- okno pro samotné hlasování s možností výběru otázky, seznamem jednotlivých vlastníků a zobrazením výsledků hlasování v reálném čase

Nejprve jsem se zaměřil na vytvoření dialogového okna sloužícího pro přidávání hlasovacího zařízení jednotlivým vlastníkům. Cílem bylo vyzkoušet vytvořený program pro BLE modul, zda pracuje správně a otestovat spojení s Bluetooth na centrálním zařízení (notebooku), jestli je možné korektně přijímat data z hlasovacího zařízení. Na konec jsem začal vytvářet okno hlavní aplikace, sloužící pro správu vlastníků, otázek a ovládání samotného hlasování.

### 5.3.1 Hlavní okno aplikace

Zde jsem narazil na první problém, jak v jednom okně přehledně zobrazit všechny potřebné informace jako jsou vlastníci, otázky a aktuální informace o probíhajícím hlasování. Měl jsem hned několik nápadů jak tohoto docílit. První nápad byl, že při spuštění aplikace by bylo zobrazeno okno pro hlasování se dvěma tlačítky, která by otvírala další dialogová okna, určená pro editaci uživatelů a otázek. Tento návrh jsem zavrhl, poté co jsem zjistil, že Qt nabízí tzv. *Stacked widget* (viz obrázek 5.3), který umožňuje mít více různých „stránek“ v jednom okně. Tento způsob mi připadá uživatelsky přívětivější než otevírání další dialogových oken. Pro přepínání mezi jednotlivými stránkami jsem nejprve chtěl využít dvojici tlačítek „předchozí a další“, jenže tento způsob se později ukázal jako neefektivní, protože uživatel může chtít přeskóčit z poslední stránky určené pro hlasování přímo na první, kde má možnost upravovat uživatele a musel by se proklikat přes stránky, které jsou mezi nimi. Proto jsem zvolil možnost, kdy mezi jednotlivými stránkami lze přecházet pomocí menu a vznikl tak základní návrh okna aplikace (viz obrázek 5.3).



Obrázek 5.3: Hlavní okno aplikace

## Správa uživatelů a otázek

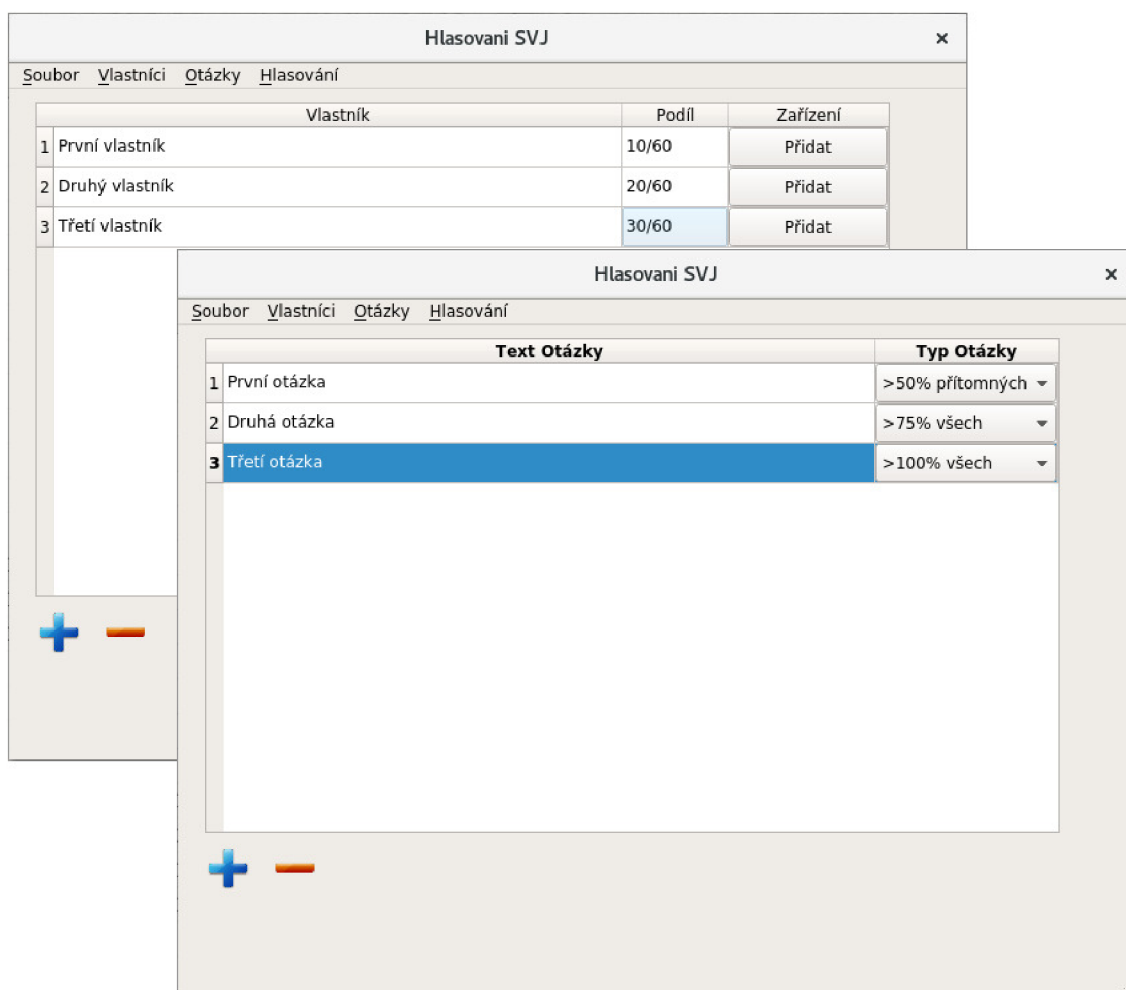
Dále jsem se zaměřil na přidávání uživatelů a otázek. Vzhled těchto dvou stránek je stejný, liší se pouze v počtu a druhu sloupců v tabulce (viz obrázek 5.4).

Tabulka pro správu uživatelů má tři sloupce. První pro jeho jméno, v druhém sloupci je uložen jeho vlastnický podíl ve tvaru zlomku (jedná se o nejčastější a nejpřesnější reprezentaci podílu) a třetí sloupec slouží pro přidání hlasovacího zařízení. Každý řádek tabulky odpovídá jedné instanci třídy „deviceInfo“, jsou zde uloženy všechny informace o vlastníkovi a jeho zařízení. Při kliknutí na tlačítko přidat uživatele je vytvořena nová instance, která je přidána do seznamu všech zařízení, v tabulce je vytvořen nový prázdný řádek, kde v posledním sloupci je zobrazeno tlačítko „přidat“. Při kliknutí na toto tlačítko je otevřeno dialogové okno (viz 5.3.2), sloužící pro přidání hlasovacího zařízení, pokud je zařízení úspěšně připojeno, místo tohoto tlačítka je zobrazen název připojeného zařízení. Uživatel může také připojené zařízení odpojit a to dvojklikem na název zařízení, po odpojení zařízení se opět zobrazí tlačítko pro připojení zařízení. Tlačítko „odstranit“ způsobí, že připojené hlasovací zařízení na vybraném řádku je nejprve odpojeno, poté je odstraněn příslušný řádek z tabulky a nakonec i instance v seznamu zařízení.

Tabulka pro správu otázek má pouze dva sloupce. První pro zadání názvu otázky a druhý pro výběr typu otázky. Podle typu otázky je zobrazován výsledek hlasování, jestli byl návrh přijat nebo ne.

Uživatel má na výběr z pěti možností:

1. více než 50 % přítomných vlastníků
2. více než 50 % všech vlastníků
3. více než 75 % přítomných vlastníků
4. více než 75 % všech vlastníků
5. 100 % všech vlastníků

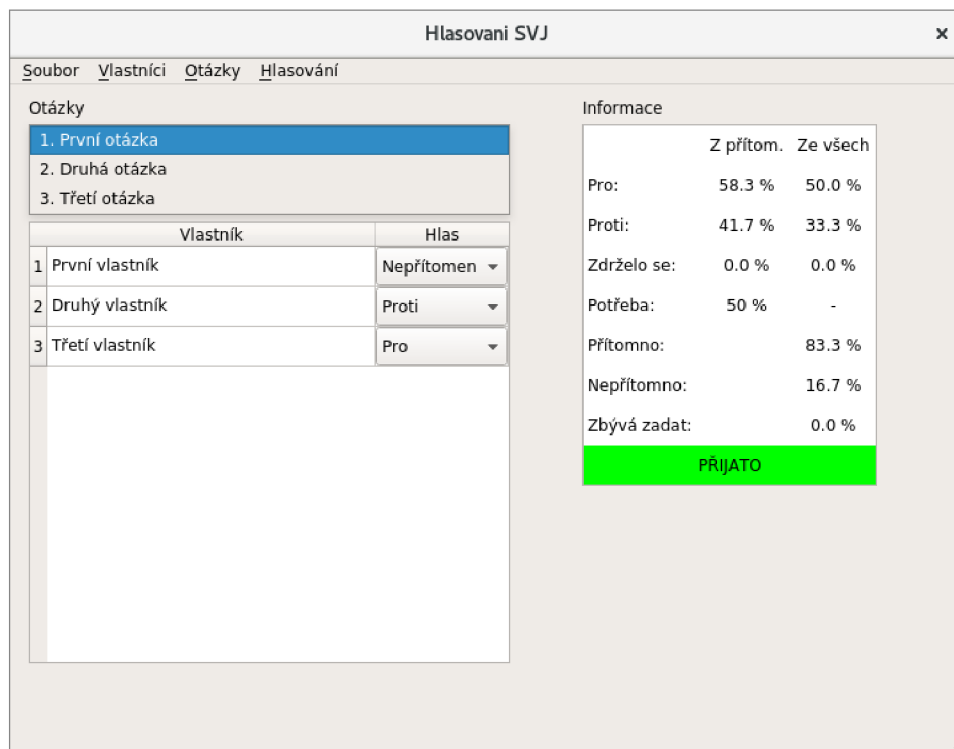


Obrázek 5.4: Vzhled stránky pro přidání uživatele a otázky

## Stránka pro hlasování

Poslední stránka slouží pro samotné hlasování. Uživatel zde má zobrazen seznam otázek, uživatelů, a také tabulku zobrazující průběh hlasování (viz obrázek 5.5). Na výběr je mezi otázkami, které byly definovány na stránce „Otázky“. Při změně otázky je upravena tabulka pro zobrazení průběžných výsledků, kde je také zobrazen typ otázky, a pro všechna připojená zařízení je nastavena defaultní hodnota zmáčknutého tlačítka. V seznamu vlastníků je

zobrazeno pouze jméno a to, jak daný vlastník hlasoval. Uživatel může hlas ručně změnit, proto je hlas zobrazen jako tzv. *combo box*, kde je kromě třech možností stisknutých tlačítek na hlasovacím zařízení také možnost, že vlastník je nepřítomen.



Obrázek 5.5: Náhled stránky pro hlasování

Aktualizace informací je prováděna každých 500 ms. Metoda pro aktualizaci informací je připojena na signál časovače, který je nastaven na danou hodnotu, vždy po vypršení intervalu je zavolána tato metoda. Metoda zajišťuje aktualizaci hodnoty stisknutého tlačítka v aplikaci, podle toho jak daný vlastník hlasoval. V případě, že vlastníkově není přiřazeno hlasovací zařízení, může uživatel hlasovat za vlastníka ručně výběrem příslušné odpovědi. Uživatel může také změnit hlas kteréhokoli vlastníka, při této změně je nová hodnota hlasu také odeslána a uložena do hlasovacího zařízení, jinak by při příštím zavolání metody byla opět nastavena původní hodnota. Tato vlastnost je přidána z důvodu jakýchkoliv problémů s hlasovacím zařízením. Pokud vlastník opustí hlasování i se svým hlasovacím zařízením a dostane se z dosahu centrálního zařízení v aplikaci se toto projeví nastavením hodnoty hlasu na „nepřítomen“. Při návratu do dosahu centrálního zařízení je vlastník opět automaticky připojen. Nevýhodou je zde, že nelze poznat, zda bylo ztraceno spojení s hlasovacím zařízením z důvodu poruchy nebo jestli vlastník odešel.

Při aktualizaci dat je také upravována tabulka pro zobrazení průběhu hlasování. Je zde zobrazeno kolik procent vlastníků hlasovalo pro, proti a kdo se zdrželo hlasování. Nejjednodušší pro výpočet jsou hodnoty pro všechny vlastníky, kde jsou pouze sečteny vlastnické podíly podle jednotlivých hlasů. Výpočet procent pouze u přítomných je již trochu složitější, protože podíly nepřítomných vlastníků jsou rovnoměrně rozděleny mezi přítomné vlastníky. Nejprve jsou tedy sečteny podíly nepřítomných vlastníků, tato hodnota je podělena počtem přítomných vlastníků a výsledná hodnota je přičtena k podílu vlastníka, takže

například vlastník, který původně měl vlastnický podíl 16/128 může mít nyní například 18/128. Hodnoty hlasů v této části jsou sice zobrazovány v procentech, ale při výpočtech nejsou hodnoty nijak zaokrouhlovány, aby bylo dosaženo co nejpřesnějšího výsledku.

Pod těmito hodnotami je zobrazen o jaký typ otázky se jedná, při otázce na přítomné vlastníky je typ napsán v levém sloupci a analogicky pro otázky na všechny vlastníky v pravém sloupci. Další dva řádky zobrazují počet přítomných a nepřítomných vlastníků. Počet nepřítomných vlastníků je velmi důležitý pro hlasování, pokud součet podílů přítomných vlastníků nepřesáhne hodnotu 50 % jedná se tzv. *neusnášeníschopnost*, v tomto stavu nelze hlasování prohlásit za platné.

Zbylé dva řádky zobrazují nijak podstatné informace pro hlasování, slouží pouze pro okamžitou zpětnou vazbu uživateli. Jedná se o počet procent vlastníků, kteří ještě nehlasovali a poslední řádek ukazuje, jestli byl návrh (otázka) přijat nebo nepřijat. Pokud je součet podílů vlastníků, kteří hlasovali „pro“ větší než součet podílů hlasů „proti“ a zároveň větší, než typ otázky (tzn. >50 %, >75 % nebo =100 %) je zobrazen text „Přiját“ na zeleném pozadí. Při opaku je zobrazen text „Nepřijat“ na červeném pozadí. Při stejném počtu pro a proti hlasů je návrh nepřijat, protože není dosaženo nadpoloviční většiny. Při poklesu hodnoty přítomných vlastníků pod 50 %, je dosaženo neusnášeníschopnosti a je zobrazen bílý text „Neusnášeníschopnost“ na černém pozadí.

### 5.3.2 Připojení hlasovacího zařízení

Vytvořil jsem nové dialogové okno, které jsem rozdělil na dvě hlavní části. V první části budou zobrazena jména nalezených zařízení, druhá část bude sloužit pro zpětnou vazbu uživateli o právě prováděné akci (viz obrázek 5.6). Okno je nastaveno jako modální, což zabraňuje provádět jakékoliv změny v jiných oknech bez zavření aktuálního nebo otevřít více aktuálních oken. V první řadě je kontrolováno jestli je Bluetooth zapnuto, pokud není, je zobrazena chybová hláška o tom, že uživatel musí zapnout Bluetooth. Pokud je vše v pořádku je automaticky spuštěno vyhledávání dostupných zařízení v blízkém okolí. Do seznamu nalezených zařízení jsou přidávána pouze ta, která jsou typu BLE, ostatní jsou přeskočena, což uživateli poskytuje přehlednější seznam připojitelných zařízení. Uživatel má možnost vyhledávání ukončit pomocí tlačítka „cancel“ nebo opakovat vyhledávání zařízení tlačítkem „scan“.



Obrázek 5.6: Dialogové okno pro připojení zařízení

Při dvojkliku na kterékoliv zařízení zobrazené v seznamu je zahájeno připojování k tomuto zařízení, ale v první řadě je zastaveno vyhledávání, pokud stále běží, protože další zařízení uživatele už nezajímají a není potřeba je vyhledávat. Jednotlivé stavy při připojování lze sledovat v pravé informační části, která může být v jednom ze čtyř stavů:

1. **Probíhá připojení k zařízení** – pokus o připojení k hlasovacímu zařízení
2. **Kontrola komunikace** – testování jestli zařízení obsahuje konkrétní službu pro čtení stisknutého tlačítka
3. **Spojení navázáno** – hlasovací zařízení bylo úspěšně připojeno k centrálnímu zařízení (např. notebooku)
4. **Chyba při připojování** – došlo k neočekávané chybě, je potřeba zkusit znovu připojit dané zařízení

Pokud připojování dosáhne stavu „Spojení navázáno“ je povoleno tlačítko pro přidání zařízení, které bylo dosud vypnuto (nereagovalo na kliknutí). Při kliknutí na toto tlačítko je aktuální okno zavřeno a veškeré informace o připojeném zařízení jsou uloženy do třídy „deviceInfo“.

## 5.4 Problémy při vývoji

První problém, na který jsem narazil při implementaci programu pro BLE modul bylo, že pokud zařízení bylo připojeno, a pak odpojeno, nebylo možné jej znovu nalézt. Zařízení se po odpojení nedostalo do stavu vysílání advertising paketu. Jelikož jsem se při vytváření programu nechal inspirovat již existujícími řešeními, nevšiml jsem si, že nemám ošetřen případ změny na BLE stacku při odpojení zařízení. Při přidání tohoto případu do funkce, která je volána při každé změně na BLE stacku byl problém vyřešen.

Další problém nastal při pokusu o implementování zápisu hodnoty. Nejprve jsem implementoval pouze čtení hodnoty, později jsem zjistil, že bude potřeba hodnotu také zapisovat.



Při volání funkce v klientské aplikaci, která měla sloužit pro ukládání hodnoty, nebyla hodnota do hlasovacího zařízení uložena. Při prostudování dokumentace a fóra firmy Cypress, jsem se dočetl, že funkce pro uložení hodnoty opět pouze změní stav na BLE stacku a obsahuje parametr s hodnotu, která má být uložena. Proto byla přidána funkce pro BLE modul, která zajistí uložení přijaté hodnoty při změně na BLE stacku.

Při vytváření aplikace pro Qt jsem se na webových stránkách dočetl o funkci, která má sloužit pro zapnutí rozhraní Bluetooth. Jenže při pokusu o zapnutí se aplikace „tvářila“, jako kdyby vše proběhlo v pořádku, nebyla vypsána žádná chybová hláška, ale Bluetooth bylo neustále vypnuto. I přes dlouhé hledání příčiny této chyby na internetu jsem nebyl schopen nalézt odpověď. Nezbyvalo tedy než tuto chybu obejít. V kódu funkce pro zapnutí Bluetooth zůstala, ale navíc se kontroluje jestli je Bluetooth zapnuto, pokud ne, je vypsána chybová hláška uživateli, aby zapnul Bluetooth zařízení manuálně. Kontrola je prováděna, až v případě, když se uživatel snaží o připojení hlasovacího zařízení k vlastníkovi, takže aplikaci lze využít i bez použití Bluetooth, ale uživatel by musel ručně zapisovat hlasy jednotlivých vlastníků.

Vznikla také komplikace při přiřazování hlasovacího zařízení vlastníkovi, protože místo tlačítka „přidat“ se měl zobrazit název připojeného zařízení, tak se také stalo, ale při přepnutí na jinou stránku a zase zpět, se opět zobrazilo tlačítko „přidat“, i když hlasovací zařízení bylo stále připojeno. Ale tlačítko nebylo zobrazeno trvale, při přidání nového řádku v tabulce, se opět zobrazil název připojeného zařízení. Problém byl nejspíše způsoben tím, že pro buňku bylo nejprve tlačítko nastaveno jako tzv. *cellWidget*, a potom byl pro buňku opět nastaven text. Řešením bylo nastavit pro buňku nový tzv. *widgetItem*, a tím odstranit nastavený *cellWidget* pro buňku. Použitím tohoto řešení bylo jméno připojeného zařízení již zobrazeno stabilně.

Největší problém, na který jsem narazil a nepodařilo se mi ho vyřešit je, že Bluetooth implementované v Qt nepodporuje desktopovou verzi Windows. Po přečtení článku na stránkách Qt (viz [15]) jsem se domníval, že když verze Qt 5.5 má podporu pro Windows 10 a přidává plnou podporu pro Bluetooth Low Energy, je možné Bluetooth využít i pod systémem Windows. Jelikož jsem aplikaci psal a testoval pouze pod linuxem na tento problém jsem narazil, až v samotném závěru. Na internetu jsem našel knihovnu *QBluetoothZero*, pomocí které by mělo být možné v Qt pracovat s Bluetooth pod Windows. Jenže podle jejich návodu se mi nepodařilo knihovnu přeložit, abych ji mohl použít. Při hledání na fóru jsem zjistil, že nejsem sám, kdo má tento problém. Při dalším hledání jsem narazil na fóru na *Qtconnectivity*, což měla být další možnost, jenže ani tato knihovna problém nedokázala vyřešit. Problém tedy zůstal nevyřešen a podle reakcí na fóru by Qt mohlo podporovat Bluetooth ve Windows ve verzi Qt 6.0. Datum vydání této verze zatím není stanoveno, uvolnění finální verze 5.9 je plánováno na 31. dubna 2017.

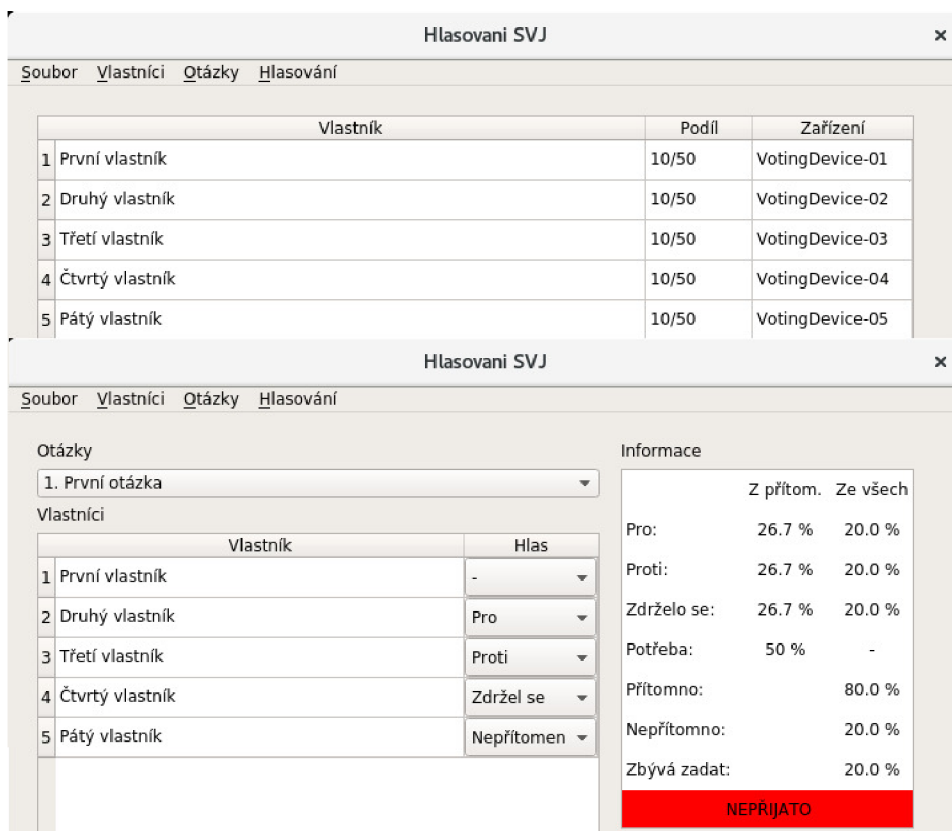
## Kapitola 6

# Ověření řešení

Tato kapitola se věnuje závěrečnému ověření funkčnosti celého systému hlasovacího zařízení, tedy jak uživatelské aplikace, tak samotného hlasovacího zařízení. U vybraných modulů bylo provedeno také ověření dosahu zařízení, které je udáváno výrobcem. Je zde také uvedeno, jaká je předpokládaná výdrž hlasovacího zařízení, pokud bude napájeno pomocí mincové baterie CR2032.

### 6.1 Ověření funkčnosti hlasování

Při hlasování je vždy přítomno hned několik vlastníků, kteří budou hlasovat, proto bylo potřeba ověřit funkčnost aplikace při zaznamenávání hlasů z několika zařízení najednou. Využity byly zakoupené PProC moduly, které jsou přímo připájeny k DPS (viz obrázek 4.1), a také mnou vytvořené hlasovací zařízení (viz obrázek 5.1). Ověření funkčnosti aplikace pro hlasování bylo prováděno za pomoci pěti připojených hlasovacích zařízení (viz obrázek 6.1). Cílem bylo ověřit, jestli aplikace dokáže zaznamenávat hlasy z několika připojených zařízení najednou, a pokud vlastník opustí dosah centrálního zařízení, tak po opětovném připojení, může opět hlasovat bez nutnosti jakéhokoliv zásahu uživatele. Opuštění dosahu centrálního zařízení bylo simulováno pomocí vypnutí a znovu zapnutí modulu. Při tomto odzkoušení aplikace nebyly zjištěny žádné chyby a všechny části aplikace splnily očekávání.



Obrázek 6.1: Konfigurace BLE zařízení

## 6.2 Ověření maximálního dosahu zařízení

Hlavní vlastností periferního zařízení, určeného pro hlasování, je jeho dosah. Udávaný dosah BLE zařízení se pohybuje okolo 30 metrů. Tato hodnota je udávána ve volném prostoru, pokud je mezi zařízeními překážka, dosah klesá. Jedinou přirozenou překážkou v hlasovací místnosti budou samotní lidé, jelikož nepředpokládám, že v hlasovací místnosti budou mezi zařízeními překážky, jako jsou zdi, apod. Proto testování vzdálenosti probíhalo ve volném prostoru, případně v prostorech, kde byl překážkou člověk. Druhé zařízení, pomocí kterého byla kontrolována komunikace s BLE zařízením, byl mobilní telefon od společnosti Sony, konkrétně model Xperia Z5.

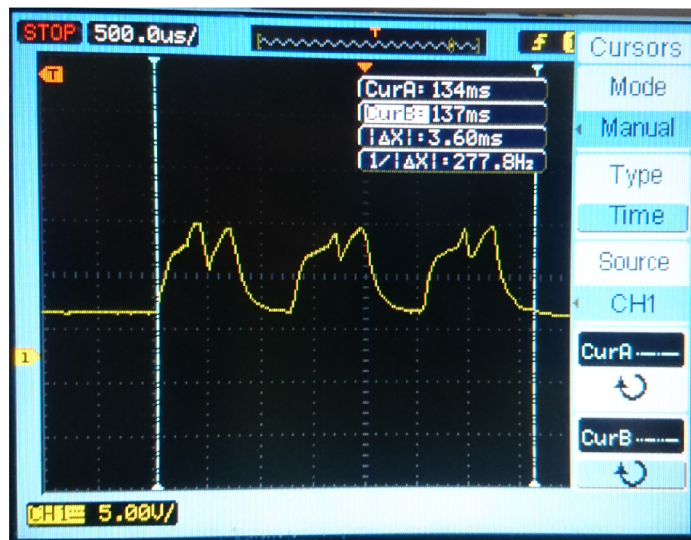
Při ověřování dosahu PSoC modulu byla testována hodnota 30 metrů, kterou udává výrobce. Maximální dosah modulu ve volném prostoru pro nalezení druhého zařízení byl 29 metrů, pro výměnu dat se dosah zvýšil dokonce na 33 metrů. Zařízení nebyly ovlivněny žádným rušením jako je wi-fi, dráty vysokého napětí, apod. Když byli mezi zařízeními postaveni dva lidé, dosah se snížil na 25 metrů.

Dosah u PSoC modulu byl ověřován na mnohem větší vzdálenosti. Výrobce udává dosah zařízení 400 metrů, nejprve je ale nutné tuto funkci zapnout. Návod, pro zapnutí této funkce lze nalézt přímo v datasheetu výrobce (viz [6]). Na tuto vzdálenost již nebylo možné zajistit spojení zařízení bez okolního rušení. Mezi zařízeními již bylo jednak hned několik wi-fi sítí, a také dráty vysokého napětí. Experimentálně změřený dosah pro nalezení zařízení byl 110 metrů, po úspěšném spojení se vzdálenost prodloužila až na 133 metrů.

### 6.3 Spotřeba periferního zařízení

Před samotným měřením na osciloskopu bylo nutné nalézt ideální odpor pro měření spotřeby. Zjistil jsem, že vhodný odpor byl  $51\ \Omega$ . Cílem měření na osciloskopu bylo zjistit spotřebu zařízení při vysílání advertising paketu a při odesílání informace o stisknutém tlačítku.

Obrázek 6.2 zobrazuje spotřebu při vysílání advertising paketu. Tři špičky spotřeby jsou způsobeny vysláním tohoto paketu na všechny tři kanály, aby byla zvýšena pravděpodobnost nalezení periferního zařízení, které má být připojeno. Pro zjednodušení jsem počítal s konstantní spotřebou po celou dobu vysílání. Při vysílání advertising paketu je z baterie spotřebováno za 1 s přibližně  $540\ \mu\text{Ah}$ .



Obrázek 6.2: Vysílání advertising paketu

U modulů byl nastaven connection interval na 100 ms, to znamená, že zařízení vysílá a přijímá data každých 100 ms. Vysílání a přijímání je vzájemně posunuto, takže zařízení ve výsledku komunikuje každých 50 ms, jak lze vidět na obrázku 6.3b. Obrázek 6.3a zobrazuje spotřebu periferního zařízení při odesílání nebo při přijímání dat. Paket sloužící pro kontrolu spojení mezi centrálním a periferním zařízením může nést až 20 B dat, tedy odesílání informace o stisknutém tlačítku se provádí právě pomocí tohoto paketu. Spotřeba periferního zařízení za 1 s se pohybuje okolo  $210\ \mu\text{Ah}$ .



(a) Odeslání dat

(b) Connection interval

Obrázek 6.3: Odesílání dat

Pro měření spotřeby při vysílání a přijímání dat byla led dioda společně s buzzerem vypnuta, proto je nutné jejich spotřebu připočítat. Zjištěná výdrž periferního zařízení na baterii je 15 dní při nepřetržitém používání pro hlasování. Výdrž ovlivňuje hlavně led dioda, která má na zařízení nejvyšší spotřebu. Pokud je zařízení připojeno nebo odpojeno, led dioda pouze bliká, ale pokud uživatel spustí hlasování, led dioda svítí a periferní zařízení čeká na stisk jednoho ze tří tlačítek. Záleží tedy na tom, jak rychle bude hlasující reagovat na začátek hlasování a na tom jak dlouho bude led dioda svítit.

## 6.4 Cena

Při výrobě prototypu byl nejdražší samotný BLE modul a výroba DPS. Výše ceny byla způsobena hlavně nízkým počtem objednaných kusů. Výrobce musí při výrobě DPS přenastavit stroj pro tuto desku, což zabere nějaký čas, proto při výrobě jednoho kusu je cena řádově vyšší, než kdyby byl požadavek vyrobit například 100 kusů. To samé platí i pro BLE modul, při větším odběru zařízení by jistě bylo možné získat množstevní cenu. V tabulce 6.1 nejsou vypsané všechny součástky použité pro výrobu prototypu, ty které jsou řádově v korunách jsou zahrnuty do položky ostatní součástky. Tabulka 6.1 ukazuje cenu pro výrobu jednoho prototypu.

Název	Cena
Výroba DPS	130 Kč
BLE modul	263 Kč
Držák baterie	30 Kč
Reproduktor	34 Kč
Ostatní součástky	43 Kč
<b>Suma:</b>	<b>500 Kč</b>

Tabulka 6.1: Cena periferního zařízení

Pro výrobu prodejního výrobku, by byla cena součástek nižší z důvodu množstevního množství (viz výše). Nutné je také připočítat položku pro automatizované osazení součástek,

protože při výrobě prototypu byly součástky osazeny ručně. Součástí by měl být také obal, umožňující přístup k baterii pro její výměnu.

# Kapitola 7

## Závěr

Cílem této bakalářské práce bylo vytvořit aplikaci, která umožňuje připojit hlasovací zařízení, komunikující pomocí BLE a následně hlasovat, s možností vložení vlastnického podílu pro jednotlivé vlastníky. Součástí bylo také vytvoření hardwarového zařízení, které bude osazeno třemi tlačítky, dvěma led diodami a buzzerem, komunikující s klientskou aplikací pomocí technologie BLE, která zaznamenává, a dále zpracovává hlasy přijaté od jednotlivých vlastníků.

Samotné implementaci předcházelo studium bezdrátové technologie BLE, způsobu navazování komunikace, profilů a služeb umožňujícími přenos dat. Popis této technologie, způsob komunikace, apod. jsou popsány v úvodních kapitolách.

Pro experimentování s touto technologií jsem využil modulů firmy Cypress, protože jejich moduly využívají energeticky úsporný 32-bitový procesor ARM Cortex-M0, a také proto, že jejich BLE moduly jsou certifikovány organizací Bluetooth SIG. Objednal jsem moduly, které jsou přímo připájeny k desce plošných spojů, aby mohly být ihned použity pro testování. Znázorněny jsou na obrázcích 4.1a a 4.1b. Zvolil jsem moduly s různým dosahem. Při ověřování dosahu PSoC modulu jsem testoval hodnotu 30 metrů, kterou udává výrobce. Při mém testování dosah zařízení pro nalezení zařízení byl 29 metrů a pro komunikaci se dosah zvýšil dokonce až na 33 metrů. U PSoC modulu, který by měl mít dosah až 400 metrů, byla experimentálně zjištěná hodnota cca 110 metrů, měření bylo ovlivněno hlavně zařízením použitým pro kontrolu spojení, a také okolním rušením.

Spotřebu periferního zařízení jsem odhadl na 15 dní při nepřetržitém provozu. Hodnota je pouze orientační, závisí hlavně na tom, jak často a jak dlouho bude svítit led dioda, sloužící pro identifikaci spuštěného hlasování. Pro implementaci klientské aplikace jsem si vybral multiplatformní framework Qt, ovšem později jsem zjistil, že to není nejvhodnější volba, protože nepodporuje práci s Bluetooth pod desktopovou verzí Windows.

Všechny cíle bakalářské práce byly splněny, kromě používání aplikace pod systémem Windows, protože multiplatformní framework prozatím nepodporuje práci s Bluetooth pod tímto systémem. Úspěšně byla navrhnutá a vyrobena deska plošného spoje sloužící jako hlasovací zařízení. Deska je osazena modulem, který má stejné vlastnosti jako zmíněný PSoC modul, jen má menší flash paměť, a to konkrétně 128 KB. Vytvořena byla také klientská aplikace ve frameworku Qt, která umožňuje shromažďovat data z jednotlivých hlasovacích zařízení a příslušným způsobem je interpretovat.

Aplikace je ve fázi prototypu a pro reálné využití při hlasování na shromáždění společenství vlastníků, je zapotřebí dodělat ještě několik dalších úprav. Aplikace nyní umožňuje zaznamenávat hlasy jednotlivých vlastníků s přepočtem váhy jejich hlasu. Aby byla použitelná pro reálné hlasování, a co nejvíce zjednodušila práci uživateli, je zapotřebí ještě

další vývoj. Při porovnání s již existující aplikací sloužící pro tento účel (viz [11]), je nutno dodělat hlavní chybějící části. Jednou z těchto částí je importování vlastníků z katastru nemovitostí, aby byla zaručena bezchybnost, a to především u domů, který má velký počet vlastníků, a také zajistit, aby byla co nejvíce ulehčena práce s přidáváním uživatelů a jejich vlastnických podílů do programu. Budoucí vývoj by měl zahrnovat také možnost tisku jednotlivých sestav, například prezenční listina, záznam o jednotlivých hlasech pro konkrétní vlastníky, apod. Potenciální vyladění nedostatků a dokončení potřebného vývoje by aplikace velmi výrazně usnadnila a urychlila průběh hlasování.



# Literatura

- [1] Bluetooth SIG: Bluetooth Low Energy. [online], 2017, [cit. 29.03.2017].  
URL <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/low-energy>
- [2] Bluetooth SIG: How It Works. [online], 2017, [cit. 27.03.2017].  
URL <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works>
- [3] Cypress Semiconductor Corporation: CYBLE-222014-EVAL EZ-BLE™ PSoC BT 4.2 Evaluation Board. [online], 2016, [cit. 08.04.2017].  
URL <http://www.cypress.com/documentation/development-kitsboards/cyble-222014-eval-ez-ble-proc-bt-42-evaluation-board>
- [4] Cypress Semiconductor Corporation: CYBLE-224110-EVAL EZ-BLE™ PSoC XT/XR Evaluation Board. [online], 2016, [cit. 08.04.2017].  
URL [http://www.cypress.com/documentation/development-kitsboards/cyble-224110-eval-ez-ble-psoc-xtxr-evaluation-board?source=search&keywords=224110&cat=software\\_tools](http://www.cypress.com/documentation/development-kitsboards/cyble-224110-eval-ez-ble-psoc-xtxr-evaluation-board?source=search&keywords=224110&cat=software_tools)
- [5] Cypress Semiconductor Corporation: EZ-BLE™, PSoC™ Bluetooth 4.2 Module. [online], 2017, [cit. 27.04.2017].  
URL <http://www.cypress.com/file/230691/download>
- [6] Cypress Semiconductor Corporation: EZ-BLE™ PSoC® XT/XR Module. [online], 2017, [cit. 27.04.2017].  
URL <http://www.cypress.com/file/230681/download>
- [7] Eugene: Getting Started with Bluetooth Low Energy. [online], 2014, [cit. 30.03.2017].  
URL <http://apprize.info/hardware/bluetooth/3.html>
- [8] Karásek, J.: Specifikace Bluetooth 4.2 je hotová: Vyšší rychlost a nižší spotřeba pro internet věci. [online], 2014, [cit. 27.03.2017].  
URL <https://smartmania.cz/specifikace-bluetooth-42-je-hotova-vyssi-rychlost-a-nizsi-spotreba-pro-internet-veci-9599/>
- [9] Laird Technologies: BLE Overview. [online], February 2013, [cit. 29.04.2017].  
URL <http://www.summitdata.com/blog/ble-overview/>
- [10] News, C. E. S.: Bluetooth Versions Walkthrough, and Bluetooth 4.0 Low Energy Development Resources. [online], June 2013, [cit. 28.03.2017].  
URL <http://www.cnx-software.com/2013/06/05/bluetooth-versions-walkthrough-and-bluetooth-4-0-low-energy-development-resources/>

- [11] Portál společenství vlastníků jednotek: Hlasování SVJ. [online], [cit. 19.04.2017].  
URL <http://www.portalsvj.cz/software/hlasovani/>
- [12] Punch Through Design: How GAP and GATT Work. [online], 2017, [cit. 30.03.2017].  
URL <https://punchthrough.com/bean/docs/guides/everything-else/how-gap-and-gatt-work/>
- [13] RF Wireless World: Bluetooth Frequency Allocations/Frequency Bands. [online], 2012, [cit. 28.03.2017].  
URL <http://www.rfwireless-world.com/Tutorials/Bluetooth-frequency-allocations.html>
- [14] SparkFun Electronics: Bluetooth Basics. [online], 2017, [cit. 28.03.2017].  
URL <https://learn.sparkfun.com/tutorials/bluetooth-basics>
- [15] The Qt Company: News & Press. [online], [cit. 19.04.2017].  
URL <https://www.qt.io/qt-news/qt-5-5-introduces-windows-10-support-adds-full-bluetooth-low-energy-official-support-for-red-hat-enterprise-linux-6-6/>
- [16] Thompson, T. J.; Kumar, C. B.; Kline, P. J.: *Bluetooth Application Programming with the Java APIs Essentials Edition*. Elsevier Science, 2008, ISBN 978-0-12-374342-8.
- [17] Townsend, K.; Davidson, R.; Cufí, C.; aj.: *Getting Started with Bluetooth Low Energy*. O'Reilly, 2014, ISBN 978-1-4919-0057-4.
- [18] Tutorial-Reports.com: Protocol Stack. [online], February 2013, [cit. 28.03.2017].  
URL <http://www.tutorial-reports.com/wireless/bluetooth/protocolstack.php>
- [19] Wikipedia: Bluetooth. [online], 27-04-2017, [cit. 27.03.2017].  
URL <https://en.wikipedia.org/wiki/Bluetooth>

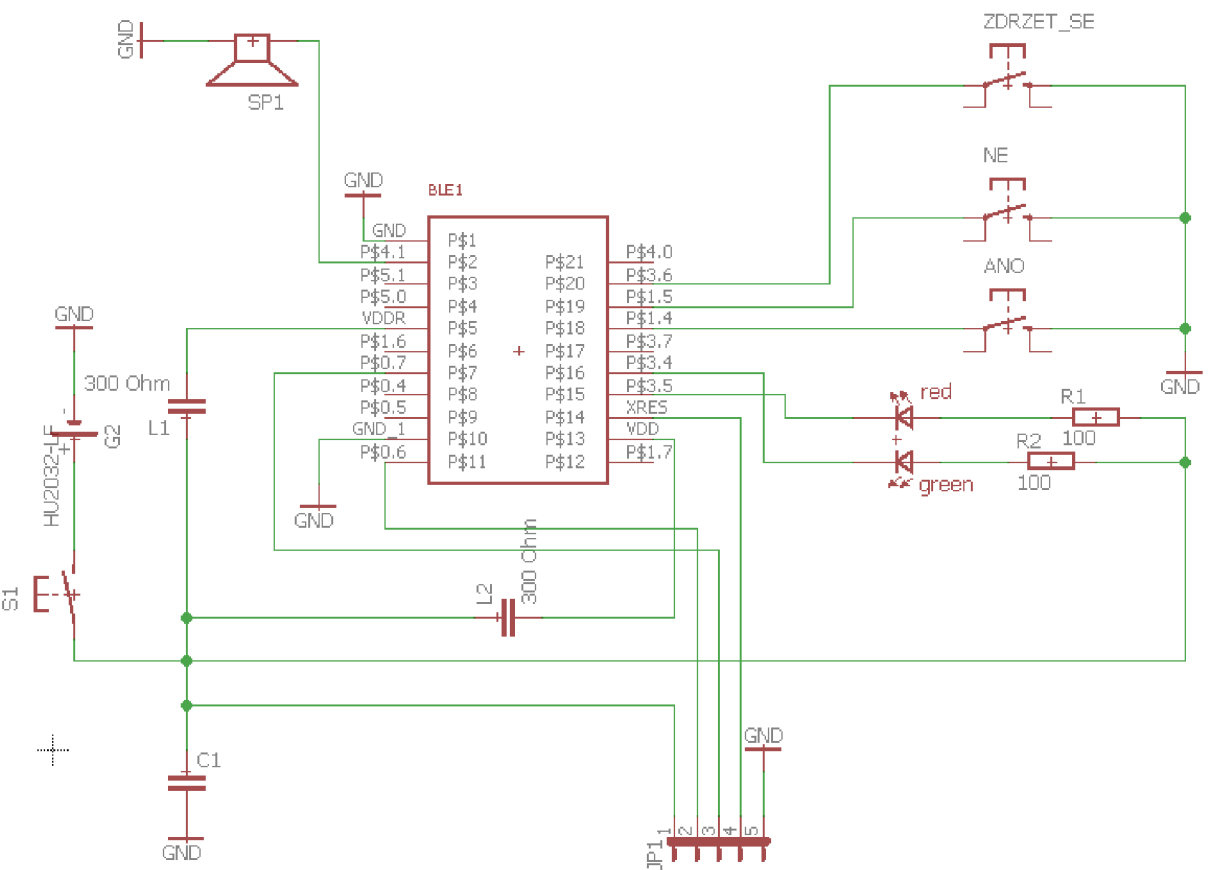
# Přílohy

## Seznam příloh

<b>A Schéma zapojení PProC modulu</b>	<b>41</b>
<b>B Obsah CD</b>	<b>42</b>

# Příloha A

## Schéma zapojení ProC modulu



# Příloha B

## Obsah CD

- Periferie – zdrojové soubory pro periferní zařízení
  - HW – soubor s návrhem DPS a schématem zapojení pro program EAGLE
  - SW – workspace pro program PSoC Creator, obsahující projekty se softwarovým řešením pro jednotlivé moduly
- Dokumentace – zdrojové soubory pro vytvoření tohoto dokumentu
- xvybir11.pdf – tento dokument