



Zadání bakalářské práce

Název:	Aplikace Android pro komunikaci s platformou Arduino protokolem Bluetooth
Student:	Andrey Denissyuk
Vedoucí:	Ing. Pavel Kubalík, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

- 1) Prozkoumejte existující řešení.
- 2) Pomocí metod softwarového inženýrství navrhnete vlastní řešení vyhovující níže uvedeným požadavkům.
- 3) Navržené řešení naprogramujte, řádně ho zdokumentujte a otestujte.

Požadavky:

- aplikace bude navržena jako mobilní aplikace pro OS Android
- aplikace umožní spojení aplikace z Arduinem pomocí Bluetooth
- pro komunikaci bude zvolen vhodný protokol
- aplikace bude umožňovat ověření uživatele s pomocí jména a hesla
- aplikace bude umožňovat přidání uživatelů s různým typem oprávnění
- aplikace umožní zobrazit několik různých pohledů (seznamu proměnných) stavu procesoru Arduina
- aplikace umožní měnit obsah proměnných v procesu Arduina

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Aplikace android pro komunikaci s platformou Arduino protokolem Bluetooth

Andrey Denissyuk

Vedoucí práce: Ing. Pavel Kubalík, Ph.D.

27. června 2021

Poděkování

Nejprve bych rád poděkoval vedoucímu mé práce Ing. Pavlovi Kubalíkovi, Ph.D., za užitečné rady a za čas, který mi věnoval během konzultačních setkání. Mé poděkování dále patří všem učitelům FIT a dalších fakult za všechny znalosti, které jsem získal během studia a které dokažu užít ve své profesi jako programátor. Rovněž bych chtěl poděkovat Taisii za morální podporu a pomoc při testování aplikace.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mé práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 27. června 2021

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2021 Andrey Denissyuk. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Denissyuk, Andrey. *Aplikace android pro komunikaci s platformou Arduino protokolem Bluetooth*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Abstrakt

Tato bakalářská práce se zabývá komunikací vývojového kitu Arduino a telefonu s operačním systémem Android pomocí Bluetooth pro obousměrný přenos dat. Dále popisuje dostupné technologie a metody pro řešení popsaného problému. Cílem práce je vytvořit mobilní aplikaci pro operační systém Android, která bude schopna přijímat data z Arduino a zobrazovat je uživateli ve vhodné formě, a také připravit desky Arduino (jak ze strany softwaru, tak i hardwaru) pro komunikaci. Výsledkem práce je prototyp mobilní aplikace a knihovna pro Arduino, která zajišťuje ukládání a přenos dat.

Klíčová slova Android, Arduino, komunikace, Bluetooth

Abstract

This bachelor thesis will examine the communication between the Arduino development kit and an Android phone using the Bluetooth technology for the two-way data transfer. It also describes available tools and methods of solving the above-mentioned problem. The objective of this thesis is to create a mobile application for the Android operating system, which will be able to receive data from the Arduino kit and display it to the end user in a suitable form, as well as preparing the Arduino board (both the software and hardware parts involved) for the communication. The result is a prototype of the mobile application and a library for the Arduino development kit, which provides storing and transferring data.

Keywords Android, Arduino, communication, Bluetooth

Obsah

Úvod	1
1 Rešerše	3
1.1 Základní požadavky	3
1.2 Existující řešení	3
1.3 Shrnutí	7
2 Analýza	9
2.1 Android	9
2.2 Arduino	11
2.3 Serializace	13
3 Návrh	17
3.1 Stanovení požadavků	17
3.2 Model případů užití	18
3.3 Diagram aktivit	19
3.4 Datový tok	21
3.5 Uživatelské rozhraní	22
3.6 Zvolené řešení	29
4 Implementace	33
4.1 Datová struktura	33
4.2 Protokol komunikace	34
4.3 Android	39
4.4 Arduino	42
5 Uživatelské testování	45
5.1 Testovací scénář	45
5.2 Příprava k testům	46
5.3 Výsledky	48

Závěr	49
Literatura	51
A Obsah přiloženého média	53

Seznam obrázků

1.1	Arduino bluetooth controller	4
1.2	Arduino Total Control free	5
1.3	Bluetooth Controller 8 Lamp	6
1.4	RemoteXY: Arduino control	7
2.1	Rozdělení operačních systémů pro mobilní telefony ve světě, upraveno podle [17].	10
2.2	Rozdělení operačních systémů pro mobilní telefony v Evropě, upraveno podle [18].	10
2.3	Rozdělení desek Arduino, upraveno podle [6].	12
3.1	Model případů užití	20
3.2	Diagram stavů vzdáleného zařízení	20
3.3	Diagram aktivit procesu komunikace	21
3.4	Datový tok	22
3.5	Hlavní stránka	23
3.6	Stránka připojení	24
3.7	Stránka přihlášení	25
3.8	Stránka přehledu proměnných	26
3.9	Stránka změny přehledů	27
3.10	Stranka přehledu obyčejné proměnné a proměnné typu readonly	28
3.11	Stranka přehledu obyčejného pole a pole typu readonly	28
3.12	Znázornění přechodů mezi jednotlivými stránkami	29
3.13	Arduino Mega 2560.	31
4.1	Diagram tříd pro Arduino	35
4.2	Diagram tříd pro Javu	36
4.3	Main Acivity	39
4.4	Connection Acivity – výběr zařízení a přihlášení	40
4.5	List Show Acivity – přehled a editace seznamu	41
4.6	Edit Variable Acivity – příklad měnných a neměnných proměnných	42

4.7	Schema připojení Bluetooth modulu	44
5.1	Schema napojení Arduino.	47

Seznam tabulek

2.1	Zastoupení verzí operačního systému Android	9
-----	---	---

Seznam zdrojových kódů

2.1	Ukázkový kód pro Arduino	13
2.2	Ukázkový kód pro XML	14
2.3	Ukázkový kód pro JSON	15
2.4	Ukázkový kód pro YAML	16
4.1	Šablona zprávy v JSON	35
4.2	Zpráva pro přihlášení	36
4.3	Odpověď s daty ve dlouhém tvaru	37
4.4	Odpověď s daty ve krátkém tvaru	38
4.5	Zpráva pro nastavení nové hodnoty proměnné	38
4.6	Ukázka práce s knihovnou <i>shared_variable.h</i>	43
5.1	Kód pro scénář - nastavení proměnných	46
5.2	Kód pro scénář - funkce setup	46
5.3	Kód pro scénář - funkce loop	46

Úvod

V současné době lidé potřebují pokročilé technologie více než kdy jindy. „Chytrá“ zařízení jednoduše činí jejich život lepší a komfortnější. Do těchto technologií není třeba investovat obrovské sumy peněz, jelikož je možné si potřebné součástky koupit a výrobek si sám vytvořit. Toto je umožněno například platformou Arduino. S její pomocí je možné vytvořit chytrý dům, meteostanici nebo domácí světlo, a to vše relativně levně. Z tohoto důvodu je dané téma důležité a relevantní jak pro studenty, tak i pro vývojáře.

Tématem bakalářské práce je vývoj aplikace na Android OS, která se dokáže připojit k platformě Arduino přes Bluetooth a v reálném čase sledovat a dálkově měnit jeho stav paměti. Díky tomu je možné číst hodnoty proměnných nebo měnit konfiguraci zařízení bez přerušení běhu programu a bez nutnosti mít počítač.

Cílem práce je vytvoření univerzální aplikace, která by plnila různé úkoly (například s různými senzory a zařízeními připojenými k Arduino). Současně by aplikace měla být dostatečně uživatelsky přívětivá, aby nemusel uživatel provádět příliš mnoho akcí.

Rešerše

V této kapitole jsou stanoveny základní požadavky budoucí mobilní aplikace. Dále je v ní uvedeno několik existujících aplikací pro operační systém Android s podobnými funkcemi, jejichž výhody a nevýhody jsou zde zanalyzovány. Následně je určeno, jak bude výsledná aplikace vypadat a co bude umět.

1.1 Základní požadavky

Nejprve je třeba zjistit, co se od aplikace očekává. Budoucí mobilní aplikace musí splňovat následující požadavky:

- aplikace se musí připojovat k vývojové desce Arduino přes Bluetooth,
- aplikace musí zobrazovat data uživateli a poskytovat možnost je měnit přes uživatelské rozhraní,
- aplikace musí rozlišovat admina a běžného uživatele,
- pro počáteční nastavení aplikace by měl být třeba minimální počet akcí.

1.2 Existující řešení

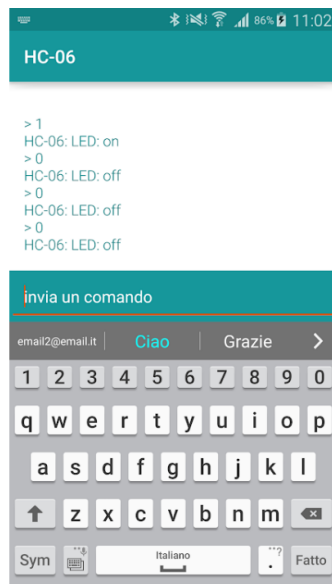
V této části jsou představeny existující aplikace s podobnými funkcemi, které jsou volně dostupné ke stažení přes Google Play [9].

1.2.1 Arduino bluetooth controller

Jedná se o aplikaci pro připojení k Arduino prostřednictvím Bluetooth a řízení projektu pomocí tlačítek a příkazového řádku [1]. Aplikace umožňuje ovládat desku Arduino pomocí terminálu, přes který může uživatel posílat

1. REŠERŠE

jednotlivé symboly, jak je uvedeno na obrázku 1.1. Také se dá použít joystick, na kterém lze konfigurovat odeslaný symbol pro každé tlačítko.



Obrázek 1.1: Arduino bluetooth controller

Výhody: Přívětivé uživatelské rozhraní, velký rozsah možností.

Nevýhody: Chybí rozlišení mezi adminem a běžným uživatelem. Délka odeslaného příkazu je pouze jeden znak, což významně omezuje možný počet příkazů.

1.2.2 Arduino Total Control free

Představuje aplikaci pro připojení k zařízením na platformě Arduino pomocí Wi-Fi nebo Bluetooth a pro posílání a příjem dat [12]. Pamatuje si použitá zařízení a umožňuje každému z nich dávat unikátní jméno a přiřkládat fotografie. Take poskytuje možnost nastavovat uživatelské rozhraní pro každé zařízení.



Obrázek 1.2: Arduino Total Control free

Výhody: Možnost rovnou se připojit k několika zařízením, možnost nastavení uživatelského rozhraní

Nevýhody: Až příliš široký rozsah možností, což může způsobit potíže nezkušenému uživateli. Opět zde chybí rozlišení mezi adminem a běžným uživatelem.

1.2.3 Bluetooth Controller 8 Lamp

Aplikace, jež se umí připojovat k Arduino a řídit osm LED diod, které zapíná a vypíná buď všechny najednou, nebo každou zvlášť [15].



Obrázek 1.3: Bluetooth Controller 8 Lamp

Výhody: Dobra ukázka práce s jednotlivým projektem na platformě Arduino, přehledné uživatelské rozhraní.

Nevýhody: Není univerzální, umí vyřešit jenom jeden konkrétní úkol, pro získání úplné verze je třeba zaplatit.

1.2.4 RemoteXY: Arduino control

„RemoteXY je snadný způsob, jak vytvořit a používat mobilní grafické uživatelské rozhraní pro desky Arduino a další řadiče. Pomocí editoru mobilních grafických rozhraní, umístěného na webu <http://remotexy.com>, můžete vytvořit své vlastní jedinečné grafické uživatelské rozhraní a načíst jej do řadiče. Pomocí této mobilní aplikace se budete moci připojit k řadiči a spravovat s nimi prostřednictvím grafického rozhraní.“ [16].



Obrázek 1.4: RemoteXY: Arduino control

Výhody: Možnost samostatné tvorby uživatelského rozhraní pro daný úkol.

Nevýhody: Není v ní naimplementovaná možnost měnit například pole z jednotlivých proměnných. Uživatelské rozhraní se nastavuje pro každý nový případ pomocí aplikace třetí strany.

1.3 Shrnutí

Jelikož je platforma Arduino velmi populární, existuje velké množství aplikací, které umožňují s ní pracovat. Nejsou však úplně univerzální, protože byly vytvořené pro konkrétní případy. Pokud aplikace slouží pro plnění velkého rozsahu úkolů, obvykle se v ní sposta věcí nastavuje ručně, což může způsobit problémy u nezkušených uživatelů. Na druhou stranu, pokud se aplikace snadno používá, nejspíš nebude příliš univerzální. Proto se autor práce bude snažit vytvořit aplikaci, jež se bude snadno používat a zároveň bude umožňovat řešit více uloh.

Analýza

Tato kapitola se zabývá operačním systémem Android a způsoby, prostřednictvím kterých je možné mobilní aplikaci vytvořit. V rámci této části práce je taktéž popsána platforma Arduino a možnosti práce s ní. Pro přenos dat mezi telefonem a touto platformou je třeba je převést do vhodné podoby, a proto jsou zde prozkoumány i nejbežnější způsoby serializace dat.

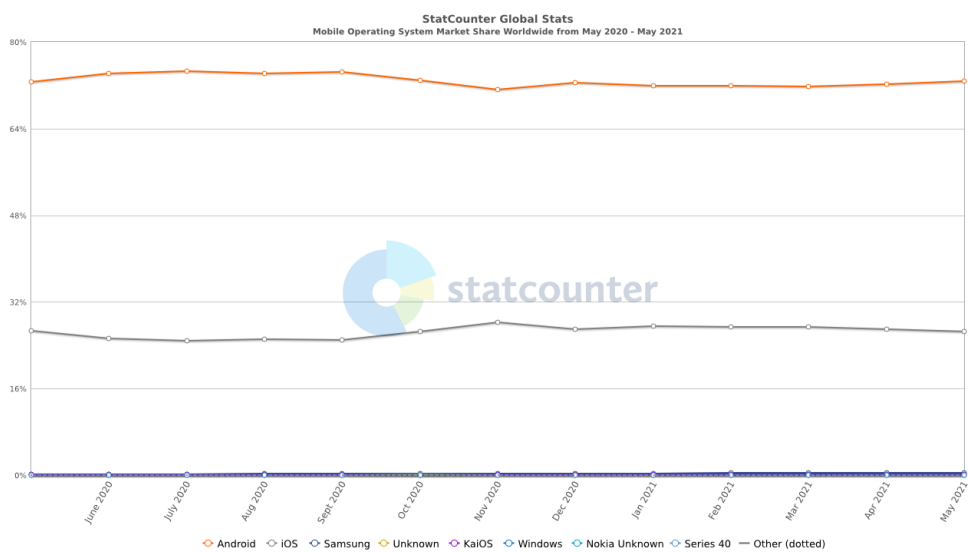
2.1 Android

Android je open source operační systém vyrobený na základě jádra Linuxu, který od roku 2008 vyvíjí společnost Google. V lednu 2021 jej používalo 71.93 % mobilních telefonů ve světě a 68.52 % těchto zařízení v Evropě, jak je vidět na obrázcích 2.1 a 2.2. Android je však součástí i chytrých hodinek, televizorů a dokonce i aut. Společnost Google neustále vydává nové verze tohoto operačního systému – seznam nepoužívanějších verzí na začátku roku 2021 je uveden v tabulce 2.1.

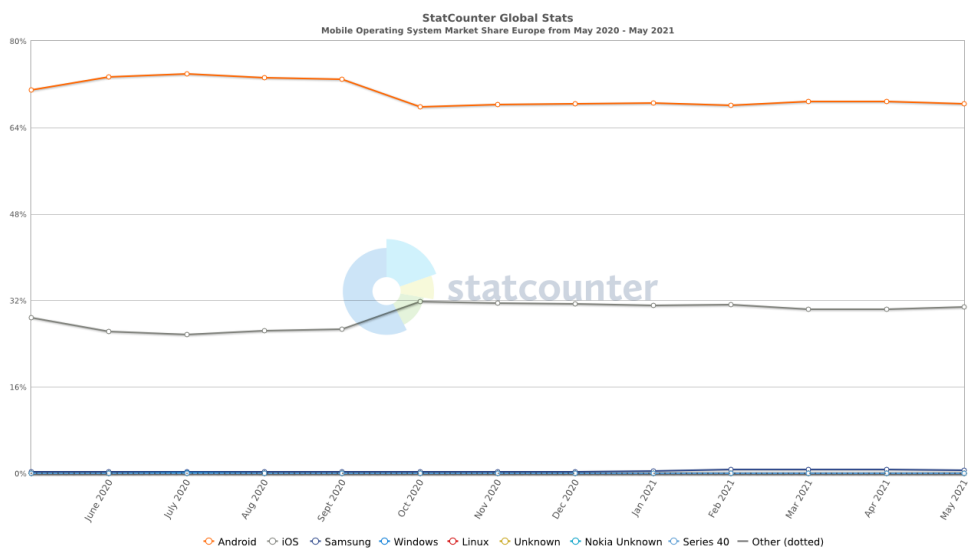
Verze	Název verze	API	Rozdělení [%]
5.0	Lollipop	21	< 1
5.1	Lollipop	22	3.16
6.0	Marshmallow	23	5.77
7.0	Nougat	24	4.09
7.1	Nougat	25	3.01
8.0	Oreo	26	5.19
8.1	Oreo	27	9.89
9.0	Pie	28	21.53
10.0	10	29	45.29

Tabulka 2.1: Zastoupení verzí operačního systému Android, upraveno podle [8]

2. ANALÝZA



Obrázek 2.1: Rozdělení operačních systémů pro mobilní telefony ve světě, upraveno podle [17].



Obrázek 2.2: Rozdělení operačních systémů pro mobilní telefony v Evropě, upraveno podle [18].

Existuje několik způsobů, jak si mohou lidé vytvořit vlastní mobilní aplikaci pro Android. Níže jsou představeny nejběžnější z nich.

2.1.1 Android Studio

Vývojové prostředí pro vytváření aplikací. Jedná se o nativní nástroj Androidu. Podporovanými programovacími jazyky jsou Java a Kotlin, přičemž jazykem pro nastavení uživatelského rozhraní je XML. Je zdarma k dispozici pro uživatele na platformách Windows, Linux a Mac OS X. Má v sobě rychlý Android emulátor, jenž usnadňuje práci programátorů a umožňuje vyvíjet aplikaci i bez mobilního telefonu [10].

2.1.2 React Native

Jedná se o framework jazyka JavaScript. Umožňuje vytváření mobilních aplikací pro operační systémy Android a iOS. V rámci něj lze používat všechny možnosti jazyka JavaScript. Komunita nástroje React Native je velká, což může začátečníkům pomoci s veškerými problémy [7].

2.1.3 Flutter

Flutter představuje open source vývojové prostředí pro mobilní aplikace vyvinuté společností Google v roce 2017. Využívá se k vývoji aplikací pro Android a iOS. Běží na vlastním enginu napsaném na C++. Používá programovací jazyk Dart [11].

2.1.4 Xamarin.Android

Xamarin.Android je framework pro vytváření mobilních aplikací pro různé operační systémy. Používá programovací jazyk C#. Je zdarma k dispozici [19].

2.2 Arduino

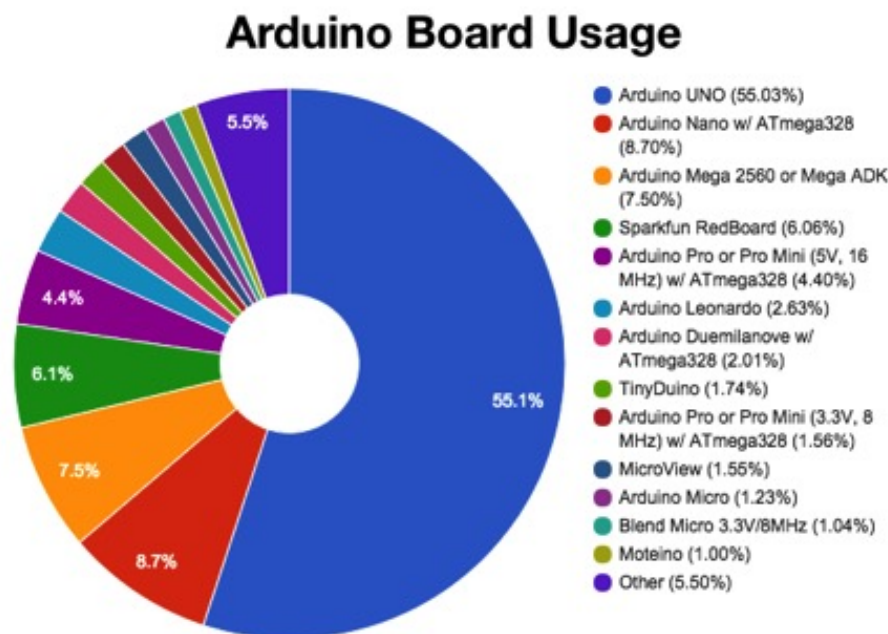
„Arduino je nástroj pro tvorbu malých počítačů, které mají větší kontrolu a cit pro fyzický svět, než běžné počítače, jak je známe. Arduino je otevřená elektronická platforma, založená na jednoduché počítačové desce (hardware) a vývojovém prostředí, které slouží k tvorbě software. Pomocí Arduina můžeme vytvářet interaktivní objekty. Arduino deska získává údaje od různých snímačů a senzorů (například snímač osvětlení, vzdálenosti nebo jen obyčejné tlačítko) a na základě těchto údajů ovládá nějaké výstupy (rozsvítí LED, zapne světlo nebo motor či jiný fyzický výstup).“ [4] Jinými slovy je Arduino univerzálním nástrojem, pomocí kterého lze vytvořit téměř cokoli přidáním periferie podle potřeb uživatelů. S programováním je možné rychle začít díky velkému množství ukázkových projektů a četné online komunitě.

V současné době patří mezi nejpopulárnější tyto vývojové desky (viz obrázek 2.3):

- Arduino UNO,

2. ANALÝZA

- Arduino Nano,
- Arduino Mega 2560/Mega ADK,
- Sparkfun RedBoard,
- Arduino Pro/Pro Mini,
- Arduino Leonardo.



Obrázek 2.3: Rozdělení desek Arduino, upraveno podle [6].

Jako vývojové prostředí se většinou používá Arduino IDE [3]. Tento program má k dispozici základní knihovny pro práci s vývojovými deskami a sadu knihoven pro spolupráci s připojeným hardwarem. Umožňuje kontrolu syntaxe kódu, nahrání programu do vývojové desky, spojení s deskou pomocí sériového monitoru atd.

Jako programovací jazyk se používá Wiring. Jedná se o framework pro jazyk C++, což znamená, že má vlastností objektově orientovaného jazyku a umí spolupracovat s hardwarem. Příklad jednoduchého programu pro desku Arduino je zobrazen v kódu 2.1.

```
int ledPin = WLED;
    // a name for the on-board LED

void setup() {
    pinMode(ledPin, OUTPUT);
        // set the pin for digital output
}

void loop() {
    digitalWrite(ledPin, HIGH);
        // turn on the LED
    delay(1000);
        // wait 1000 milliseconds
    digitalWrite(ledPin, LOW);
        // turn off the LED
    delay(1000);
        // wait 1000 milliseconds
}
```

Listing 2.1: Ukázkový kód pro Arduino

2.3 Serializace

Serializace je proces převodu datové struktury do formátu vhodného pro přenos dat anebo pro uložení do souboru. Existuje mnoho způsobů serializace dat, přičemž níže jsou uvedeny ty nejběžnější.

2.3.1 XML

XML (Extensible Markup Language) je jednoduchý a velmi flexibilní textový formát, který umožňuje definovat vlastní tagy a atributy. Tento jazyk se označuje jako rozšiřitelný, protože neopravuje značku použitou v dokumentech – vývojář si ji může libovolně vytvořit podle vlastností konkrétní oblasti. Schopnost vytvářet vlastní značky dělá XML univerzální.

```
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Listing 2.2: Ukázkový kód pro XML

2.3.2 JSON

JSON (Java Script Object Notation) je lehký formát pro výměnu dat mezi počítači, jenž byl odvozen z literatury Java Script. JSON je kompaktnější než XML, jeho konstrukce lze jednodušeji analyzovat pomocí Java Script, pro který je JSON použitým interním typem dat. Hlavní oblastí působnosti JSON je programování webových aplikací, kde slouží jako alternativa k XML.


```
{
  "widget" : {
    "debug" : "on",
    "window" : {
      "title" : "SampleKonfabulatorWidget",
      "name" : "main_window",
      "width" : 500,
      "height" : 500
    },
    "image" : {
      "src" : "Images/Sun.png",
      "name" : "sun1",
      "hOffset" : 250,
      "vOffset" : 250,
      "alignment" : "center"
    },
    "text" : {
      "data" : "ClickHere",
      "size" : 36,
      "style" : "bold",
      "name" : "text1",
      "hOffset" : 250,
      "vOffset" : 100,
      "alignment" : "center"
    }
  }
}
```

Listing 2.3: Ukázkový kód pro JSON

2.3.3 YAML

YAML je formát pro serializaci dat, který je čitelný pro lidi a koncepčně blízký značkovacím jazykům. Je však zaměřený na pohodlí vstupů a výstupů pro typické datové struktury v mnoha programovacích jazycích. V raných fázích byl považován za konkurenta XML, nicméně později se zaměřil na práci s daty namísto značkování.

2. ANALÝZA

```
name: Martin D'vloper
job: Developer
skill: Elite
employed: True
foods:
  - Apple
  - Orange
  - Strawberry
  - Mango
languages:
  perl: Elite
  python: Elite
  pascal: Lame
education: |
  4 GCSEs
  3 A-Levels
  BSc in the Internet of Things
```

Listing 2.4: Ukázkový kód pro YAML

Návrh

V této kapitole jsou stanoveny požadavky na konečnou aplikaci a následně je v ní vytvořen model případů užití a diagram aktivit. Dále je zde navržen model datového toku a uživatelské rozhraní. Na konci kapitoly jsou určeny vybrané programy a zařízení pro řešení daného problému.

3.1 Stanovení požadavků

Nejprve je třeba definovat požadavky, což zjednoduší další práci na projektu a zajistí, že na konci provede program přesně to, co od něj bude požadováno.

3.1.1 Funkční požadavky

Funkční požadavky popisují funkce systému. Seznam těchto požadavků pro aplikaci je následující:

- **F1 – Vyhledání dostupných zařízení**
Aplikace bude umožňovat vyhledávání dostupných vzdálených zařízení a zobrazování jejich seznamu.
- **F2 – Bluetooth spojení**
Aplikace bude poskytovat možnost komunikovat pomocí technologie Bluetooth. Aplikace bude umožňovat přijímat a odesílat data.
- **F3 – Zobrazení dat**
Aplikace bude ukazovat uživateli seznam sdílených proměnných a obsah každé proměnné.
- **F4 – Změnění dat**
Aplikace bude umožňovat uživateli měnit obsah sdílených proměnných, pokud to nebude zakázáno.

- **F5 – Různé uživatele**
Aplikace bude podporovat několik uživatelů s různými oprávněními.
- **F6 – Přihlášení**
Uživatel se bude muset do aplikace přihlásit prostřednictvím PIN kódu.
- **F7 – Různé přehledy**
Aplikace bude mít několik přehledů (seznamů) sdílených proměnných.
- **F8 – Nastavení přehledů**
Aplikace bude umožňovat uživateli nastavovat přehledy sdílených proměnných.
- **F9 – Ukládání přehledů**
Konfigurace přehledu se bude ukládat do paměti telefonu.
- **F10 – Aktualizování dat**
Aplikace bude neustále aktualizovat data během komunikace.

3.1.2 Nefunkční požadavky

Nefunkční požadavky představují omezení pro daný systém. Specifikují způsoby implementace a definují podmínky pro systém, avšak nesouvisí s uživatelským používáním systému. Jedná se o následující požadavky:

- **N1 – Jazyk**
Aplikace bude napsána v Javě.
- **N2 – Format dat**
Aplikace a Arduino budou přenášet data ve formátu JSON.
- **N3 – Protokol komunikace**
Data budou přenašena pomocí vlastního protokolu.
- **N4 – Role při komunikaci**
Během komunikace se aplikace bude chovat jako klient a Arduino jako server.

3.2 Model případů užití

Níže je uveden popis jednotlivých případů užití:

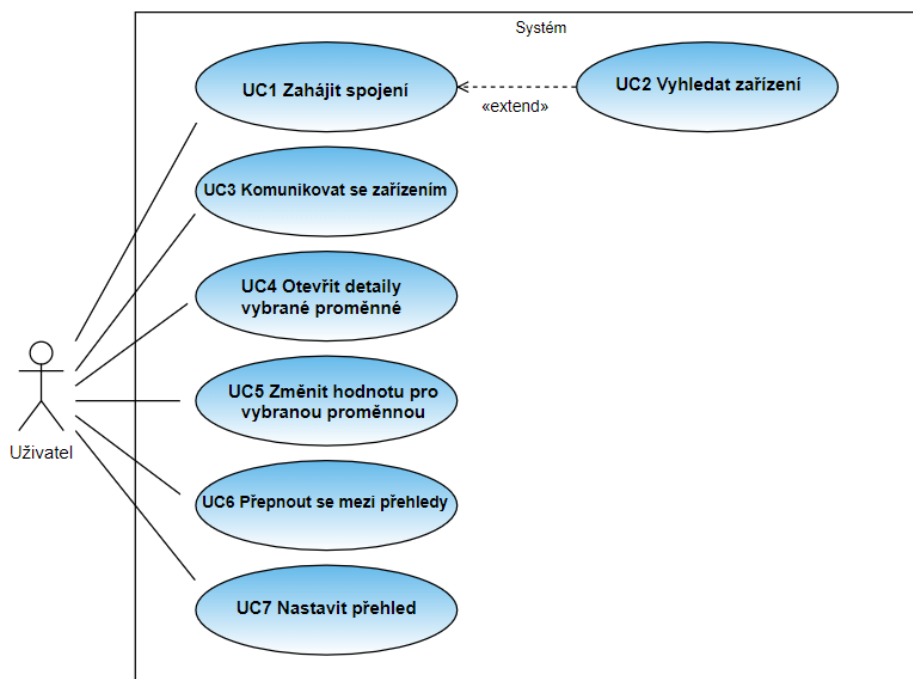
- **UC1 – Zahájit spojení**
Systém umožní uživateli navázat spojení s vybraným vzdáleným zařízením.

- **UC2 – Vyhledat zařízení**
Systém umožní uživateli vyhledat vzdálená zařízení v dosahu, aby mohl s některým z nich navázat spojení.
- **UC3 – Komunikovat se zařízením**
Systém umožní uživateli komunikovat se zařízením, se kterým navázal spojení.
- **UC4 – Otevřít detaily vybrané proměnné**
Systém umožní uživateli otevřít detaily vybrané proměnné, ukáže její hodnotu a další vlastnosti.
- **UC5 – Změnit hodnotu pro vybranou proměnnou**
Systém umožní uživateli změnit hodnotu pro vybranou proměnnou, pokud to nebude zakázáno.
- **UC6 – Přepnout se mezi přehledy**
Systém umožní uživateli přepínat mezi přehledy proměnných.
- **UC7 – Nastavit přehled**
Systém umožní uživateli nastavit přehled, což znamená změnit název nebo seznam vypsaných proměnných.

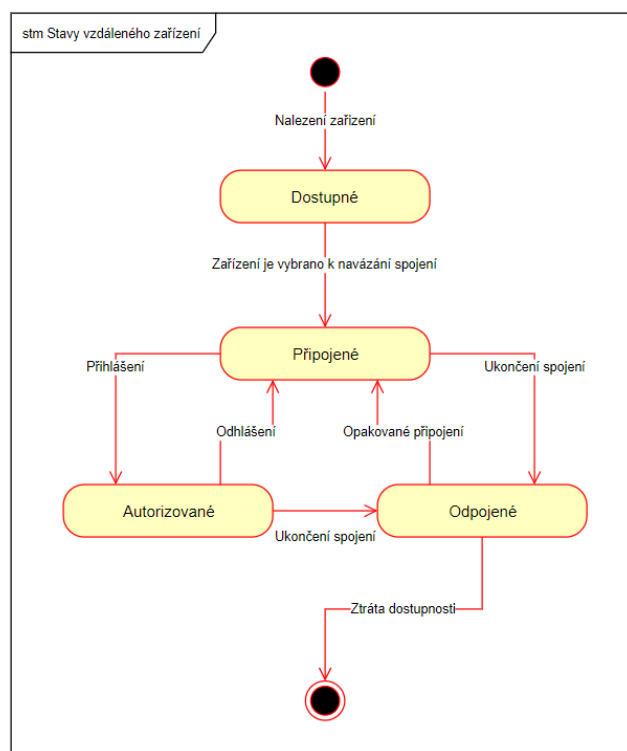
3.3 Diagram aktivit

Hlavním obchodním procesem aplikace je komunikace mezi zařízeními. Diagram aktivit pro tento proces lze vidět na diagramu 3.3. Vstupním objektem v procesu navázání komunikace je vzdálené zařízení. Změny stavů vzdáleného zařízení lze vidět na stavovém diagramu 3.2.

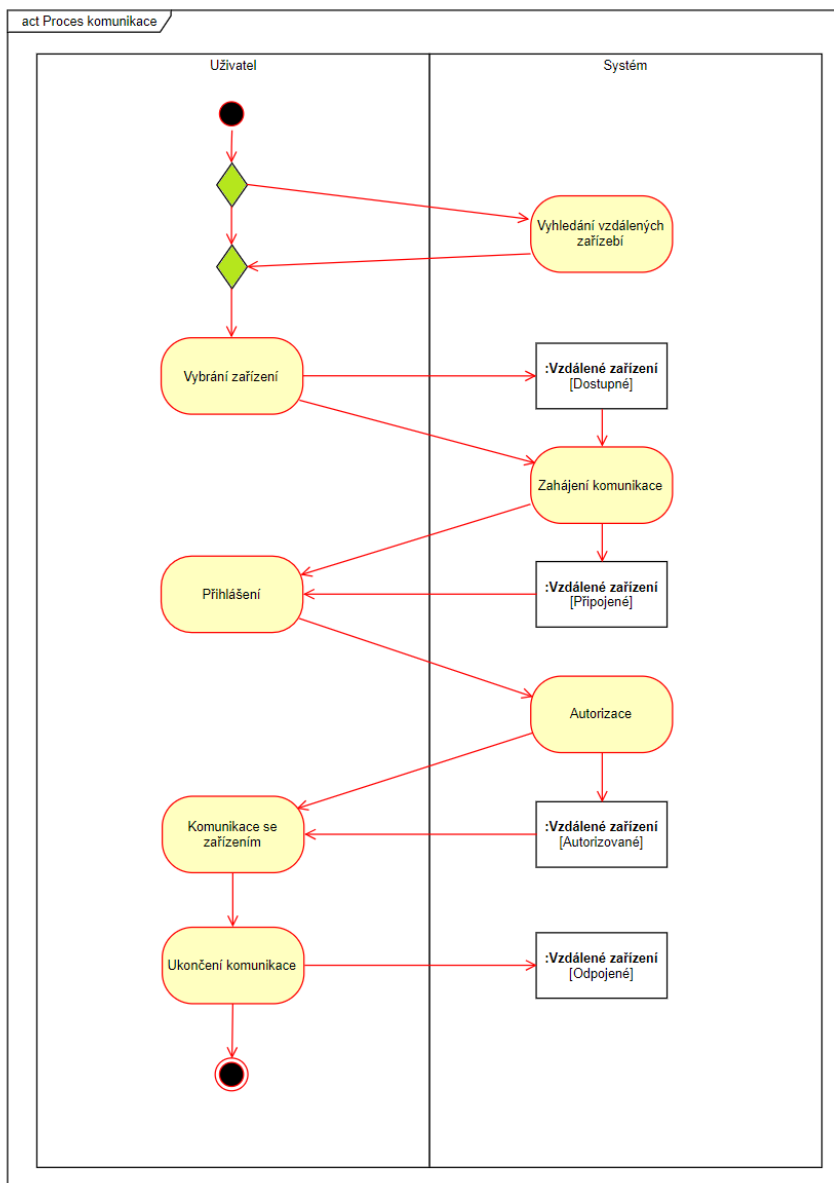
3. NÁVRH



Obrázek 3.1: Model případů užití



Obrázek 3.2: Diagram stavů vzdáleného zařízení



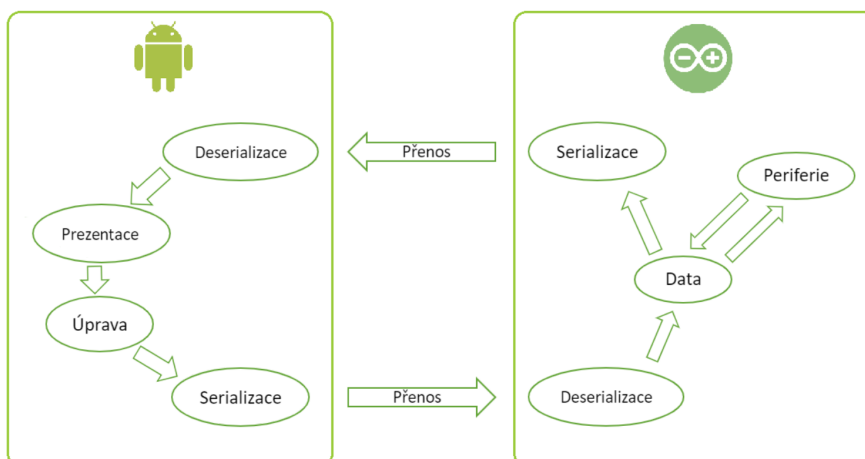
Obrázek 3.3: Diagram aktivit procesu komunikace

3.4 Datový tok

Na obrázku 3.4 je uvedeno schéma toku dat mezi aplikací a vývojovou deskou. Data se připravují na straně Arduina, poté se serializují a přenášejí do aplikace. Aplikace tato data deserializuje a zpracovává a následně je zobrazí

3. NÁVRH

uživateli. Uživatel může provést jejich změnu přes UI, potom se nová data zpátky přenesou do Arduina a zaktualizují se. Ty kroky se během aktivního připojení neustále opakují.



Obrázek 3.4: Datový tok

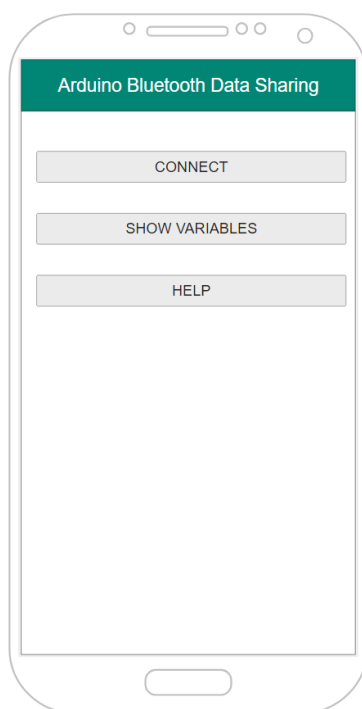
3.5 Uživatelské rozhraní

Aplikace se bude skládat ze čtyř základních stránek:

- hlavní stránka,
- stránka připojení,
- stránka přihlášení,
- stránka přehledu a editace seznamu proměnných,
- stránka přehledu a editace obsahu zvolené proměnné.

3.5.1 Hlavní stránka

Z této stránky se uživatel dostane buď na stránku připojení, na stránku přehledu proměnných anebo na pomocnou stránku. Pokud se v budoucnosti do aplikace přidají další stránky (např. nastavení, výběr jazyka, výběr způsobu připojení), budou také dostupné z hlavní stránky.

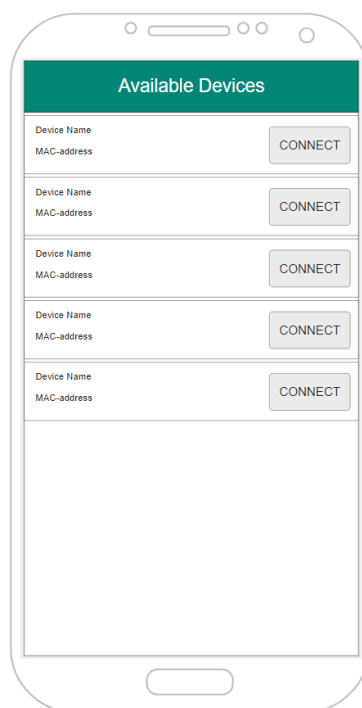


Obrázek 3.5: Hlavní stránka

3.5.2 Stránka připojení

Jedná se o stránku pro zahájení připojení. Zobrazuje uživateli seznam spárovaných zařízení a pro každé zařízení vypisuje jméno a MAC-adresu.

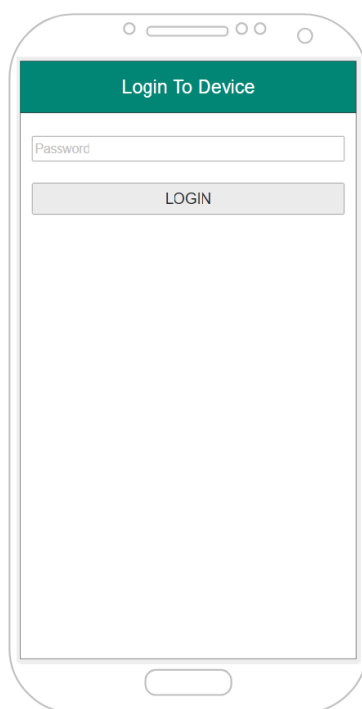
3. NÁVRH



Obrázek 3.6: Stránka připojení

3.5.3 Stránka přihlášení

Jednoduchá stránka přihlášení pomocí hesla.

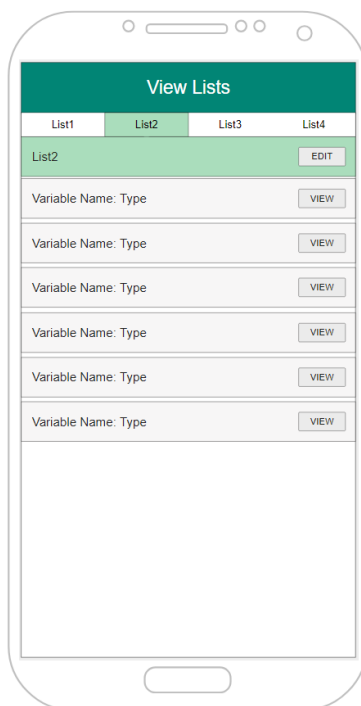


Obrázek 3.7: Stránka přihlášení

3.5.4 Stránka přehledu proměnných

Tato stránka zobrazuje čtyři seznamy proměnných, přičemž u každého se dá změnit název a proměnná, které obsahuje. Uživateli je umožněno přepínat mezi jednotlivými seznamy pomocí tlačítek v horní části stránky. Pro každou proměnnou vypisuje její název a typ a nabízí tlačítko pro otevření detailů proměnné.

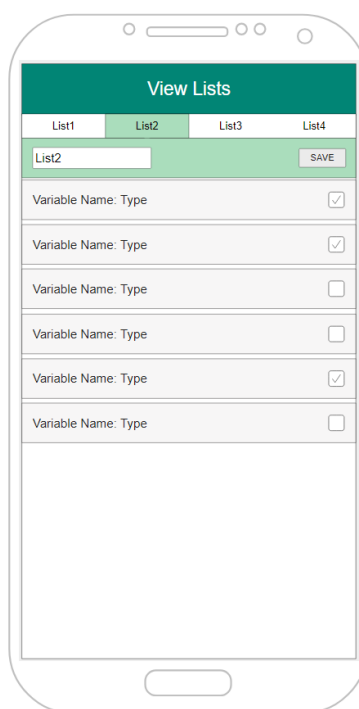
3. NÁVRH



Obrázek 3.8: Stránka přehledu proměnných

3.5.5 Stránka změny přehledů

Tato stránka umožňuje uživateli měnit jména a obsah seznamů proměnných.

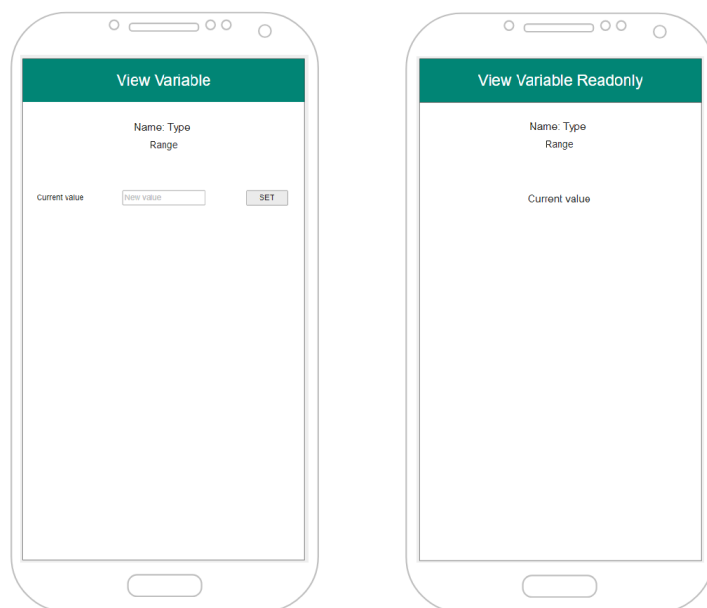


Obrázek 3.9: Stránka změny přehledů

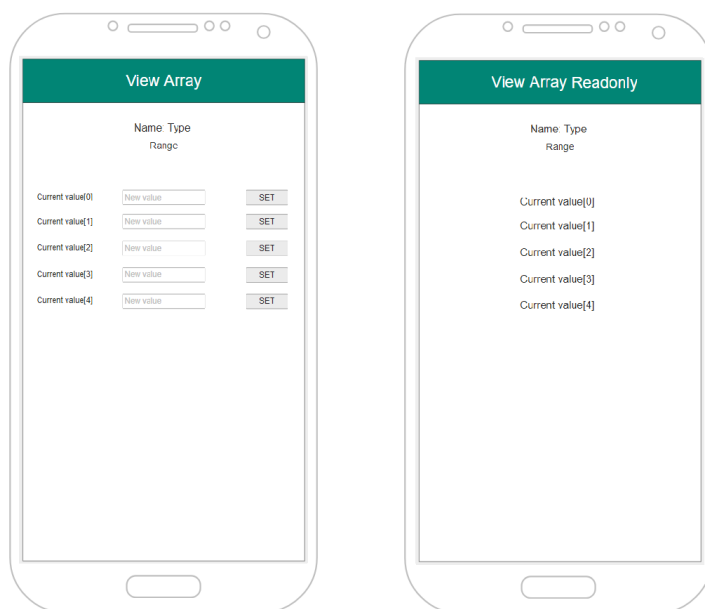
3.5.6 Stránka přehledu a editace proměnné

Tato stránka zobrazuje všechny detaily proměnné: název, typ, hodnota a rozsah možných hodnot. Pokud není zakázáno tuto proměnnou měnit, obsahuje textové pole a tlačítko pro nastavení nové hodnoty

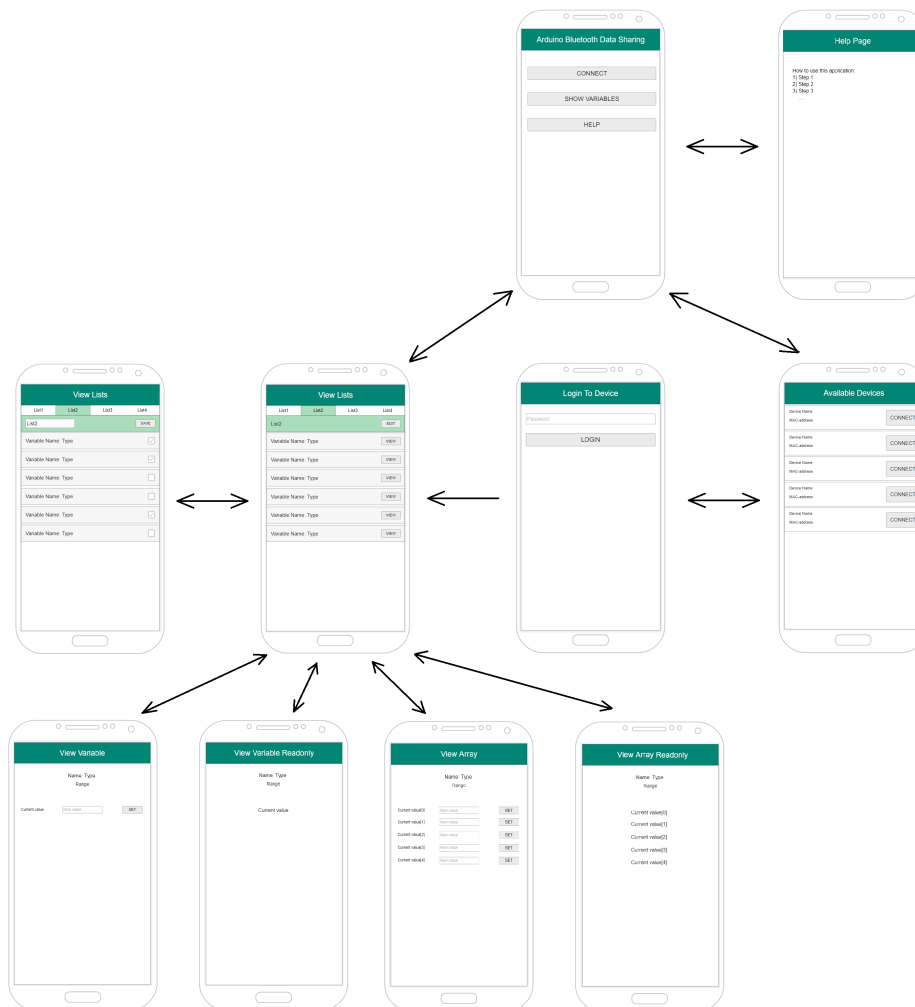
3. NÁVRH



Obrázek 3.10: Stranka přehledu obyčejné proměnné a proměnné typu readonly



Obrázek 3.11: Stranka přehledu obyčejného pole a pole typu readonly



Obrázek 3.12: Znázornění přechodů mezi jednotlivými stránkami

3.6 Zvolené řešení

Na základě provedené analýzy a vytvořeného návrhu byly zvoleny následující programy, nástroje a knihovny.

3.6.1 Metoda serializace

Jako metoda serializace byl zvolen formát JSON, protože v současné době je nejbežnějším způsobem pro vývoj webových technologií. Může se však pou-

žít i v jiných oblastech. Existuje pro něj hodně knihoven skoro pro všechny běžně používané programovací jazyky [13].

3.6.2 Android

Vývojové prostředí

Jako vývojové prostředí pro mobilní aplikaci bylo vybráno Android Studio, jelikož má hodně společného s prostředím IntelliJ IDEA, se kterým se autor práce seznámil během studia. Umožňuje pouštět aplikaci na emulátoru v reálném čase a má množství dalších vlastností hodných pro vývoj.

Programovací jazyk

Mobilní aplikace se bude psát v Javě, a to z toho důvodu, že se jedná o nativní jazyk Androidu

Knihovna pro práci s JSON

Pro práci s formátem JSON v Javě byla zvolena knihovna JSON-java [14], jelikož je pro toto použití vhodná a její kód se čte snadně a intuitivně.

3.6.3 Arduino

Vývojová deska

Jako vývojová deska byla zvolena Arduino Mega 2560, obrázek 3.13. Tato deska má následující charakteristiky:

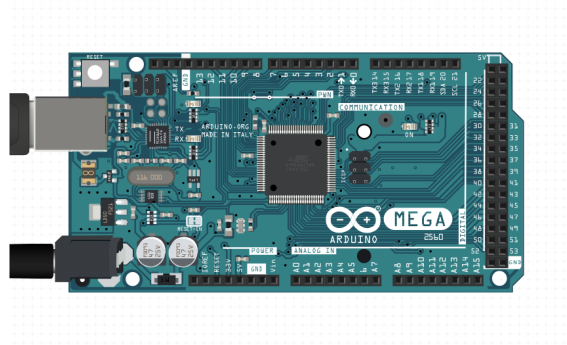
- Flash Memory – 256 KB,
- SRAM – 8 KB,
- EEPROM – 4 KB,
- Clock Speed – 16 MHz,
- Digital I/O Pins – 54,
- Analog Input Pins – 16 [2].

Bluetooth modul

z důvodu jeho popularity byl vybrán bluetooth modul HC-05. Tato volba byla rovněž ovlivněna tím, že na internetu lze najít mnoho příruček zaměřujících se na jeho připojení k vývojové desce a jeho správné nastavení.

Knihovna pro práci s JSON

Pro práci s formátem JSON na straně platformy Arduino byla zvolena knihovna ArduinoJson [5], protože je vhodná pro práci s hardwarem a nemá problémy s kompatibilitou oproti knihovnám pro obyčejný jazyk C++.



Obrázek 3.13: Arduino Mega 2560.

Implementace

V této kapitole je popsáno řešení daného problému, které je v ní rozdělené na menší části. Je zde ukázána implementace datové struktury pro usnadnění práce s daty a poté popsán realizovaný protokol komunikace. Rovněž je zde představena implementaci mobilní aplikace a knihovny pro platformu Arduino.

4.1 Datová struktura

Jak bylo uvedeno výše, jako metoda serializace dat byl zvolen formát JSON. Bylo však také nutné definovat formát, ve kterém budou data přenášena. Pro správnou funkčnost a zobrazení dat v aplikaci musí tedy JSON-object pro jednu proměnnou obsahovat následující pole:

- **Id**
Unikátní identifikátor proměnné.
- **Name**
Název, který se zobrazí v mobilní aplikaci.
- **Type**
Typ proměnné.
- **Value**
Aktuální hodnota proměnné.
- **Min**
Minimální možná hodnota pro proměnnou (pouze pro numerické typy).
- **Max**
Maximální možná hodnota pro proměnnou (pouze pro numerické typy).
- **Size**
Pokud proměnná je pole, tato hodnota je délka pole.

- **IsImmutable**

Pokud nastaveno na true, nedá se změnit hodnotu této proměnné.

- **IsSecured**

Pokud nastaveno na true, tato proměnná je viditelná pouze pro uživatele, který je přihlašen jako admin. Jinak je viditelná pro všechny přihlášené uživatele.

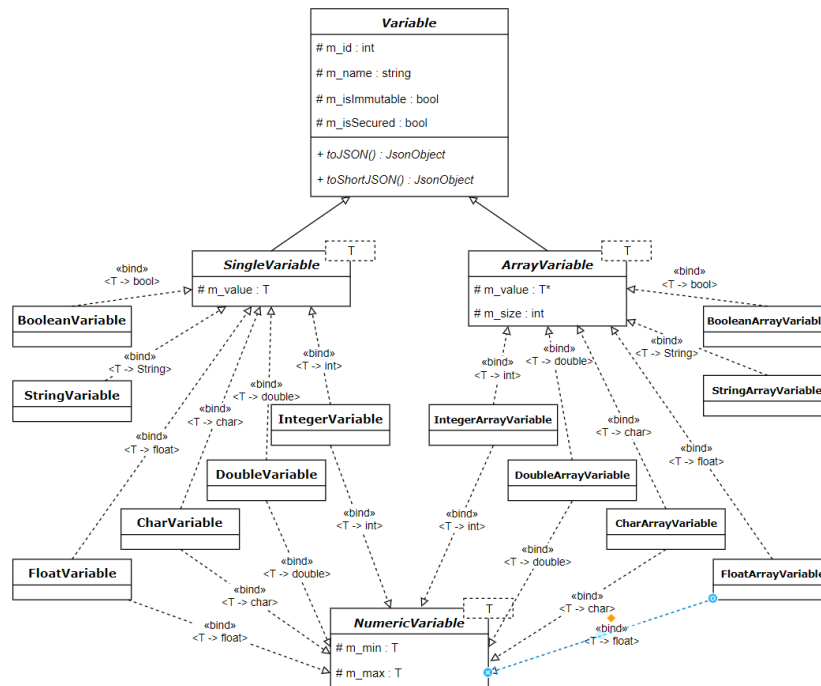
Aplikace podporuje práci s několika základními typy dat:

- int,
- float,
- double,
- char,
- bool,
- String,
- int[],
- float[],
- double[],
- char[],
- bool[],
- String[].

Pro každý podporovaný typ dat je implementována třída jak na straně aplikace, tak i na straně platformy Arduino. Pro usnadnění práce a vyhnutí se opakování kódu je použito třídního dědění, jak je vidět na obrázku 4.1. Jelikož jazyk Java nepovoluje dědění do několika tříd, byly třídy opraveny (viz obrázek 4.2).

4.2 Protokol komunikace

Veškeré zprávy v obou směrech mají univerzální tvar, jak je patrné z obrázku 4.1.



Obrázek 4.1: Diagram tříd pro Arduino

```

{
  "message type" : "...",
  "data" : {
  }
}

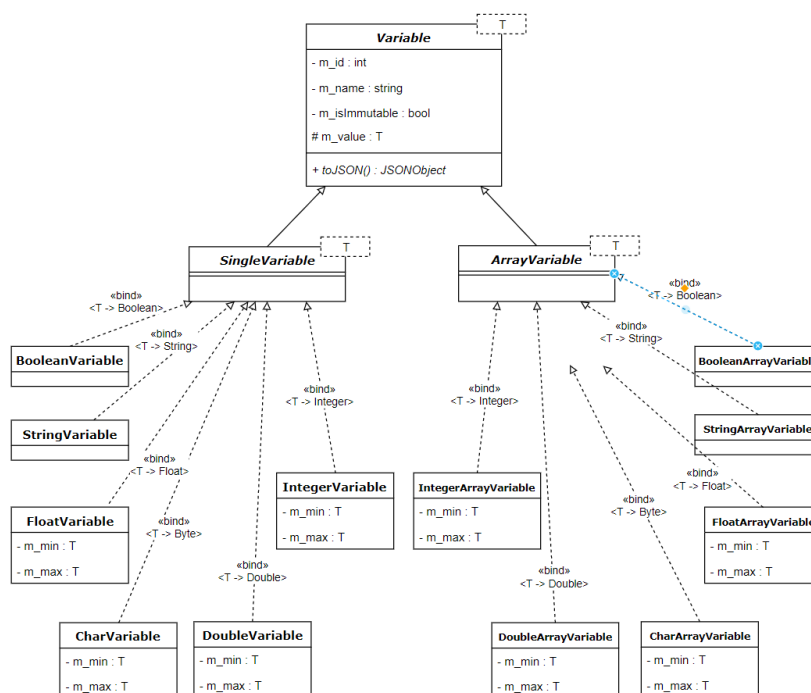
```

Listing 4.1: Šablona zprávy v JSON

Pole **message type** je hlavičkou zprávy, **data** je její tělem.

4.2.1 Login

Na začátku komunikace očekává Arduino zprávu pro přihlášení ve tvaru uvedeném na obrázku 4.2.



Obrázek 4.2: Diagram tříd pro Javu

```

{
  "message type" : "LOGIN",
  "data" : {
    "pass" : "1234"
  }
}

```

Listing 4.2: Zpráva pro přihlášení

Pokud je zadané heslo správné, pošle platforma Arduino odpověď s hlavičkou **LOGIN_OK** a bude schopna posílat data. Pokud je heslo špatné, odpoví zprávou s hlavičkou **LOGIN_FAILED**. Bude-li hlavička zprávy odlišná od **LOGIN**, pošle Arduino odpověď s hlavičkou **BAD_REQUEST**. Uživatel je považován za přihlášeného pouze tehdy, když zadá správné heslo.

4.2.2 Data import

Po úspěšném přihlášení může uživatel dostávat data od Arduina. K tomu slouží dva typy hlaviček:

- **SEND_ALL**

Uživatel posílá zprávu s hlavičkou **SEND_ALL** a dostává odpověď s hlavičkou **ALL_DATA** a úplným popisem všech proměnných. Používá se pro počáteční konfiguraci aplikace. Příklad odpovědi je uveden v kódu 4.3.

```
{
  "message type" : "ALL_DATA",
  "data" : {
    "variables" : [
      {
        "id": 0,
        "name" : "int_var",
        "type" : "int",
        "min" : -1023,
        "max" : 1023,
        "value" : 42,
        "isImmutable" : false,
        "isSecured" : false
      },
      {
        "id": 1,
        "name" : "bool_var",
        "type" : "boolean",
        "value" : true,
        "isImmutable" : false,
        "isSecured" : false
      }
    ]
  }
}
```

Listing 4.3: Odpověď s daty ve dlouhém tvaru

- **SEND_ALL_SHORT**

Uživatel posílá zprávu s hlavičkou **SEND_ALL_SHORT** a dostává odpověď s hlavičkou **ALL_DATA_SHORT** a krátkým popisem všech proměnných, součástí kterého je jenom ID a hodnota. Používá se pro aktualizaci dat na straně aplikace. Posílá se automaticky z aplikace každých 10 sekund. Příklad odpovědi je možné vidět v kódu 4.4.

```
{
  "message type" : "ALL_DATA_SHORT",
  "data" : {
    "variables" : [
      {
        "id": 0,
        "value" : 42
      },
      {
        "id": 1,
        "value" : true
      }
    ]
  }
}
```

Listing 4.4: Odpověď s daty ve krátkém tvaru

4.2.3 Data update

Pro nahrávání změn dat z aplikace do Arduina se používají zprávy s hlavičkou **UPDATE_DATA**. Tělo zprávy tedy obsahuje ID proměnné a její novou hodnotu pro nastavení na straně platformy Arduino. Příklad odpovědi je uveden v kódu 4.5.

```
{
  "message type" : "UPDATE_DATA",
  "data" : {
    "id": 0,
    "value" : 42
  }
}
```

Listing 4.5: Zpráva pro nastavení nové hodnoty proměnné

Následně Arduino nastaví hodnotu pro proměnnou s daným ID a pošle zpátky zprávu s hlavičkou **UPDATE_DATA_OK**.

4.2.4 Logout

Pokud Arduino nedostává žádné zprávy během 60 sekund (což by nemělo při normálním běhu programu nastat), automaticky ukončí komunikaci. Uživatel to tudíž nemusí dělat ručně – stačí, aby mobilní aplikaci zavřel.

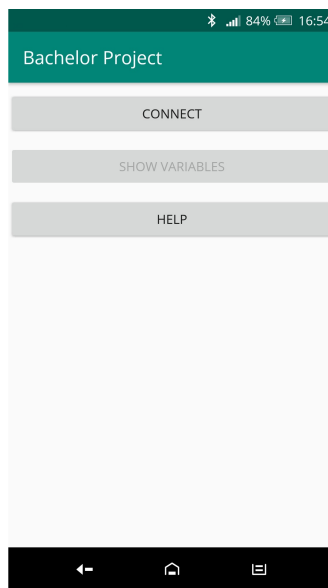
4.3 Android

4.3.1 Android Activities

Pro zjednodušení vývoje a rozdělení funkčnosti je aplikace rozdělena na několik stránek (aktivit). Níže je vypsán seznam těchto aktivit s jejich úplným popisem.

- **MainActivity**

Hlavní stránka, kterou vidí uživatel až po otevření aplikace (viz obrázek 4.3). Má tlačítka **Connect** (pro připojení k Arduino), **Show Variables** (pro otevření stránky se sdílenými proměnnými) a **Help** (pro zobrazení uživatelské příručky).

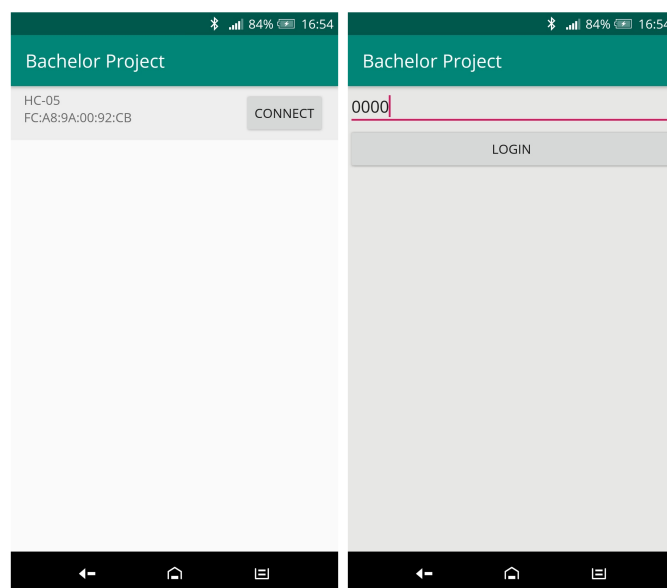


Obrázek 4.3: Main Activity

- **ConnectionActivity**

Stránka, která uživateli ukazuje seznam spárovaných zařízení, jež jsou v blízkosti telefonu (viz obrázek 4.4). Uživatel si zvolí zařízení a připojí se k němu stisknutím tlačítka **Connect** vedle názvu zařízení. Poté se otevře okno přihlášení, kde uživatel musí zadat své heslo a přihlásit se.

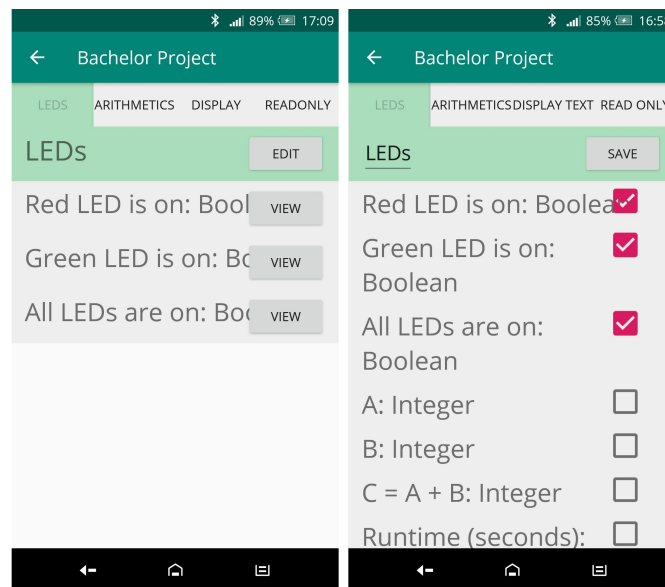
4. IMPLEMENTACE



Obrázek 4.4: Connection Activity – výběr zařízení a přihlášení

- **ListsShowActivity**

Tato stránka nabízí seznam sdílených proměnných (viz obrázek 4.5.). Je zde několik přehledů, mezi kterými může uživatel přepínat. Přehled se dá přejmenovat a jeho obsah změnit, což znamená ukázat, jaké proměnné bude ukazovat. Přehled představuje seznam proměnných, přičemž každá má takový název, který byl zadán na straně Arduina, a je zde uveden i její typ. Tlačítkem **Edit** se uživatel dostane na stránky editace vybrané proměnné.

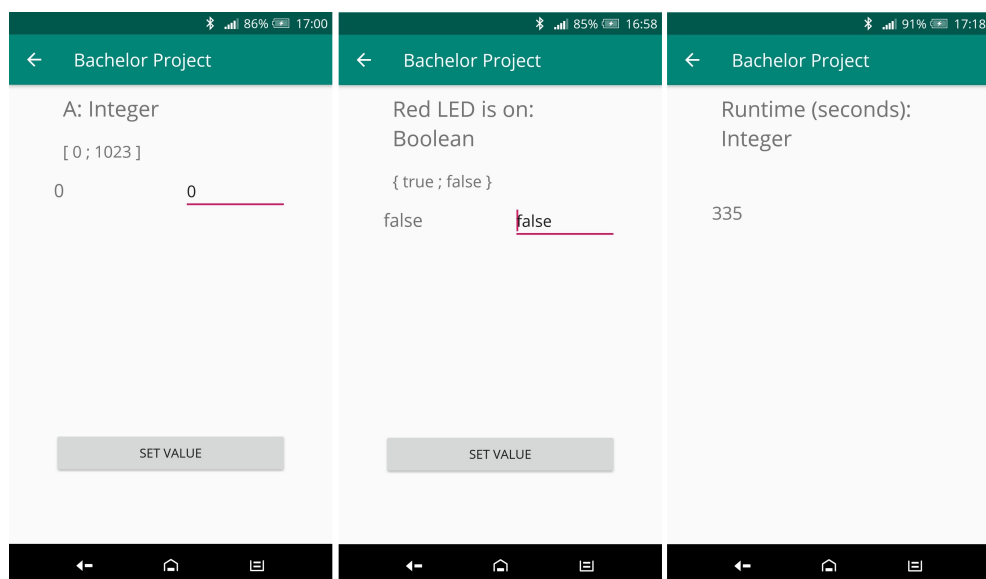


Obrázek 4.5: List Show Activity – přehled a editace seznamu

- **EditVariableActivity**

Stranka editace proměnné. Obsahuje veškeré dostupné informace jako její název, hodnota, případně i minimální a maximální možné hodnoty (viz obrázek 4.6). Pokud je povoleno měnit hodnotu proměnné, uživatel zadá její novou hodnotu a klikne na **Set Value**.

4. IMPLEMENTACE



Obrázek 4.6: Edit Variable Activity – příklad měnných a neměnných proměnných

4.3.2 Komunikace

Pro komunikaci aplikace pomocí Bluetooth je implementovaná třída **BT-ConnectionThread**. Po zahájení připojení se spouští vlákno, které přijímá a posílá data přes `BluetoothSocket`.

4.3.3 Práce s daty

Kolekce proměnných je v aplikaci uložena jako globální statický objekt, protože má být z různých tříd a aktivit. Přehledy jsou také předsaveny jako objekty. Pro serializaci a deserilizaci jsou naimplmentované třídy `JSONSerializer` a `JSONParser`, které používají knihovnu **org.json**. Pokaždé, když objekt **BTConnectionThread** přijímá zprávu, parsuje nejprve její hlavičku pak reaguje podle typu zprávy.

4.4 Arduino

4.4.1 Knihovna

Pro tento projekt byla na straně platformy Arduina naimplementována knihovna s pracovním názvem „shared_variable“. Knihovna zajišťuje komunikaci přes sériové porty, snadné ukládání dat a jejich serializaci, deserilizaci a přenos. Jelikož nepracuje přímo s modulem Bluetooth, je schopná zajistit

komunikaci i jinými způsoby (přes USB, Wi-Fi atd.). Knihovna je naimplementována tak, aby byla zmenšena velikost kódu, který bude muset uživatel přidávat do svého programu. Ukázka jednoduchého programu, který dovoluje aplikaci zapínat a vypínat LED diodu, je uvedena v kódu 4.6.

Listing 4.6: Ukázka práce s knihovnou *shared_variable.h*

Tento řádek inicializuje připojení.

```
Connection connection = Connection();
```

Následně se inicializuje sdílená proměnná a nastavuje se její hodnota, název a případně další parametry. Knihovna vytvoří objekt pro tuto proměnnou, přidá ji do vlastní kolekce a vrátí referenci na členskou proměnnou odpovídající za hodnotu.

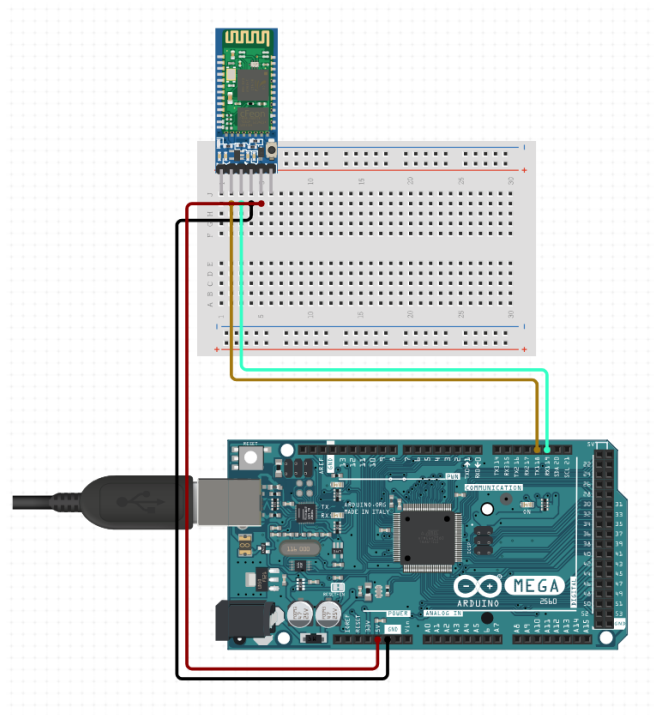
4.4.2 Hardware

Modul Bluetooth byl připojen k desce podle obrázku 4.7. Nic jiného se pro komunikaci nepotřebuje.

4.4.3 Práce s daty

Objekty popisující proměnné se ukládají do společné kolekce uvnitř knihovny. Kolekce je instancí vlastní třídy `Variable Collection`, která se zabývá uklánáním objektů proměnných. Serializace a deserializace probíhá pomocí knihovny **ArduinoJson**.

4. IMPLEMENTACE



Obrázek 4.7: Schema připojení Bluetooth modulu

Uživatelské testování

Pro pochopení toho, jak jednoduché a uživatelsky přívětivé je uživatelské rozhraní výsledné mobilní aplikace, bylo nutné provést uživatelské testování.

Za tímto účelem byl vyvinut scénář, který byl navržen pro několik lidí:

- **Uživatel A** – má dobré zkušenosti s programováním a prací s platformou Arduino,
- **Uživatel B** – má základní programovací zkušenosti, není obeznámen s platformou Arduino
- **Uživatel C** – není obeznámen s programováním, nemá žádné zkušenosti s platformou Arduino.

Po ukončení testování byla od každého uživatele získána zpětná vazba ohledně toho:

- jestli se mu práce s aplikací líbila,
- jestli pro něj bylo uživatelské rozhraní srozumitelné,
- jestli měl během testu s něčím problémy.

5.1 Testovací scénář

Scénář pro testování se skládal z několika jednoduchých kroků:

- otevření aplikace,
- připojení k desce Arduino a přihlášení pomocí hesla,
- otevření okna se seznamy dostupných proměnných,
- zapnutí červené LED diody,

Listing 5.1: Kód pro scénář - nastavení proměnných

- vypnutí červené LED diody,
- zapnutí zelené LED diody,
- vypnutí zelené LED diody,
- zapnutí obou LED diod najednou,
- zjištění doby běhu platformy,
- přečtení tajné zprávy,
- změna textu na displeji,
- nastavení hodnoty proměnné „A“ na 23,
- nastavení hodnoty proměnné „B“ na 77,
- čekání několik sekund a otevření detailu proměnné „C = A + B“,
- ověření toho, zda se hodnota této proměnné nastavila na součet A a B.

5.2 Příprava k testům

5.2.1 Arduino

Pro ten test k vývojové desce byly připojeny následující prvky (viz 5.1):

- bluetooth modul **HC-50**,
- červená a zelená LED diody,
- **OLED I2C** display.

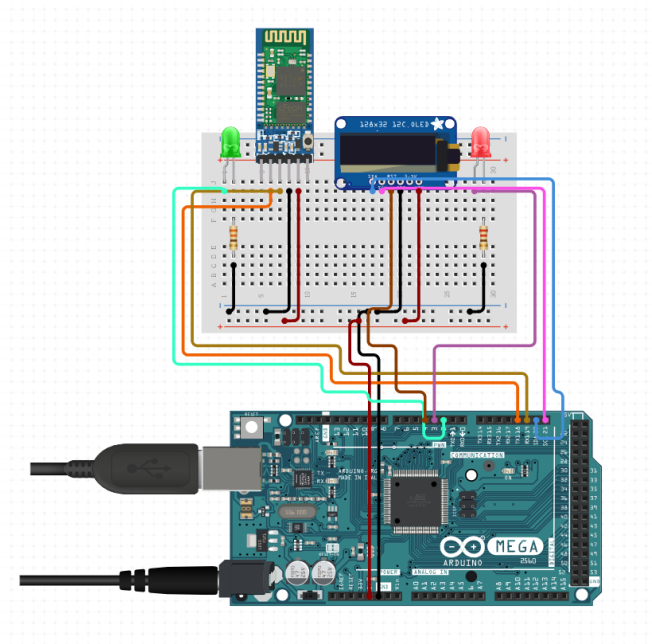
A také byl nahran kód pro práci s připojenou periferií.

Listing 5.2: Kód pro scénář - funkce **setup**

Listing 5.3: Kód pro scénář - funkce **loop**

Popis proměnných:

- **red_led_on** – ovládá červenou LED diodu;
- **green_led_on** – ovládá zelenou LED diodu;



Obrázek 5.1: Schema napojení Arduino.

- `all_leds_on` – ovládá obě LED diody;
- `int_varA` – první operand součtu;
- `int_varB` – druhý operand součtu;
- `int_varC` – výsledek součtu;
- `run_time_secs` – doba běhu programu v sekundách;
- `secret_message` – tajná zpráva;
- `text_on_display` – text na displeji.

5.2.2 Aplikace

V mobilní aplikaci byly ručně nastaveny přehledy proměnných následujícím způsobem:

- **LEDs:**
 - `red_led_on`,
 - `green_led_on`,

- all_leds_on.

- **Arithmethics:**

- A,
- B,
- $C = A + B$.

- **Display text:**

- text_on_display.

- **Read only:**

- secret_message,
- run_time_secs.

5.3 Výsledky

Prostřednictvím testování aplikace uživateli byly odhaleny následující problémy:

- **Dlouhá doba čekání na aktualizaci dat**

Aplikace získávala aktuální data z platformy Arduino každých 10 sekund. Tato doba byla snížena na 5 sekund.

- **Nedostatky uživatelského rozhraní**

Tlačítko **Set Value** bylo přesunuto nahoru vedle pole hodnoty.

Obecné dojmy z aplikace byly pozitivní. Uživatelské rozhraní vypadalo velmi jednoduše a intuitivně.

Závěr

Cílem bakalářské práce bylo vytvoření mobilní aplikace, která bude umožňovat komunikaci s vývojovou deskou Arduino přes Bluetooth. Aplikace musela být schopna přijímat data a měnit je v reálném čase. Aplikace také měla být univerzální a snadno použitelná.

Ve výsledku byla vytvořena aplikace, která komunikuje s platformou Arduino pomocí příkazů formátu JSON, díky čemuž je komunikační protokol jednoduchý a umožňuje jeho další rozšiřování. Aplikace umožňuje zobrazovat data uživateli v jasné formě a usnadňuje jejich změnu. Aplikace nevyžaduje počáteční konfiguraci, uživatel tudíž nemusí přesně vědět, jaká data Arduino obsahuje.

Pro tento úkol byla napsána knihovna pro Arduino, která připravuje data pro přenos a zjednodušuje psaní kódu. Tuto knihovnu lze rozšířit o práci s jinými datovými typy implementací odpovídajících tříd. Knihovnu lze použít i pro jiné komunikační metody (např. Wi-Fi a USB). Knihovna „shared_variable“ téměř nepoužívá knihovny třetích stran (s výjimkou ArduinoJson pro práci s JSON), proto zabírá velmi málo místa v paměti. Díky tomu je možné tuto knihovnu využít na deskách, které mají menší paměť. Mobilní aplikace byla testována a funguje přesně tak, jak má.

Na závěr je uveden seznam možných vylepšení pro budoucí vývoj aplikací:

- přidání více podporovaných typů (architektura to snadno umožní),
- doplnění možnosti vybrat jazyk uživatelského rozhraní,
- zlepšení uživatelského rozhraní,
- přidání možnosti uložit dříve použitá zařízení.

Literatura

- [1] APPS, G.: *Arduino bluetooth controller*. [online], Google, Inc., 2012, cit. 2021-03-01. Dostupné z: <https://play.google.com/store/apps/details?id=com.giumig.apps.bluetoothserialmonitor>
- [2] ARDUINO.CC: ARDUINO MEGA 2560 REV3. [online], Arduino, 2019, cit 2021-06-08. Dostupné z: <https://store.arduino.cc/mega-2560-r3>
- [3] BLANCHON, B.: *ArduinoJson*. [online], github.com, 2014, cit. 2021-03-07. Dostupné z: <https://github.com/bblanchon/ArduinoJson/>
- [4] ELECTRONICS, S.: The Arduino popularity contest. In: *www.sparkfun.com*, [online], SparkFun Electronics, 2015, cit 2021-06-08. Dostupné z: <https://www.sparkfun.com/news/1982>
- [5] FACEBOOK: *React Native*. [online], Facebook, Inc., 2020, cit. 2021-06-13. Dostupné z: <https://reactnative.dev/>
- [6] GOOGLE: *Google Play*. [online], Google, Inc., 2012, cit. 2021-03-01. Dostupné z: <https://play.google.com/store/apps>
- [7] GOOGLE: *Android Studio*. [software], Google, Inc., 2013, cit. 2021-05-15. Dostupné z: <https://developer.android.com/studio/>
- [8] GOOGLE: *Flutter*. [online], Google, Inc., 2017, cit. 2021-06-26. Dostupné z: <https://flutter.dev/>
- [9] HWKITCHEN.CZ: *Arduino IDE*. [online], hwkitchen.cz, 2014, 2014, cit. 2021-06-26. Dostupné z: <https://bastlirna.hwkitchen.cz/arduino-ide/>
- [10] HWKITCHEN.CZ: *Co to je Arduino?* [online], hwkitchen.cz, 2014, 2014, cit. 2021-06-26. Dostupné z: <https://bastlirna.hwkitchen.cz/co-je-to-arduino/>

- [11] INTEGRATIONS, A.: *Arduino Total Control free*. [online], APKPure, 2016, cit. 2021-03-01. Dostupné z: <https://apkpure.com/arduino-total-control-free/com.apps.emim.btrelaycontrolfree>
- [12] JSON.ORG: *Introducing JSON*. [online], json.org, 2020, cit. 2021-03-07. Dostupné z: <https://json.org/>
- [13] LEARY, S.: *JSON-java*. [online], github.com, 2015, cit. 2021-03-07. Dostupné z: <https://github.com/stleary/JSON-java/>
- [14] MERAH KEMARUN: *Bluetooth Controller 8 Lamp*. [online], Google, Inc., 2012, cit. 2021-03-01. Dostupné z: <https://play.google.com/store/apps/details?id=dev.merahkemarun.btcontrollamp>
- [15] SHEMANUEV: *RemoteXY: Arduino control*. [online], Google, Inc., 2012, cit. 2021-06-14, překlad vlastní. Dostupné z: <https://play.google.com/store/apps/details?id=com.shevauto.remotexy.free>
- [16] STATCOUNTER: Browser Market Share Europe. [online], StatCounter, cit. 2021-06-10. Dostupné z: <http://gs.statcounter.com/os-market-share/mobile/europe>
- [17] STATCOUNTER: Browser Market Share Worldwide. [online], StatCounter, cit. 2021-06-10. Dostupné z: <http://gs.statcounter.com/os-market-share/mobile/worldwide>
- [18] STATISTA.COM: Platform Versions. In: *statista.com*, [online], Statista, cit. 2021-02-13. Dostupné z: <https://www.statista.com/statistics/921152/mobile-android-version-share-worldwide/>
- [19] XAMARIN: *Xamarin.Android*. [online], Microsoft, Inc., 2011, cit. 2021-04-28. Dostupné z: <https://visualstudio.microsoft.com/xamarin/>

Obsah přiloženého média

```
/
├── readme.txt.....stručný popis obsahu média
├── arduino-bluetooth-data-sharing.apk..... spustitelná aplikace
├── src
│   ├── Arduino
│   │   ├── ConnectionTester
│   │   │   ├── ConnectionTester.ino..... zdrojový kód aplikace
│   │   │   └── libs
│   │   │       ├── ArduinoJson.h.....knihovna pro práci s JSON
│   │   │       └── shared_variable.h.....implementace knihovny
│   ├── BachelorProject... kořenový adresář se zdrojovými kódy aplikace
│   ├── app..... modul aplikace
│   ├── Thesis ..... zdrojová forma práce ve formátu LATEX
│   │   ├── BP_DENISSYUK_ANDREY_2021.tex . hlavní zdrojový soubor práce
│   └── BP_DENISSYUK_ANDREY_2021.pdf ..... text práce ve formátu PDF
```