



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra radioelektroniky

## Mobilní aplikace pro ovládání laboratorního napájecího zdroje

Bakalářská práce

**Michaela Jurková**

Vedoucí práce: Ing. Bc. Marek Neruda, Ph.D.

Obor: Multimediální technika

Studijní program: Komunikace, multimédia a elektronika

Květen 2017

## Poděkování

Děkuji vedoucímu práce Ing. Bc. Markovi Nerudovi, Ph.D a Ing. Bc. Lukášovi Vojtěchovi, Ph.D za cenné rady a vstřícný přístup při tvorbě bakalářské práce.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 26. května 2017

.....

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Jurková** Jméno: **Michaela** Osobní číslo: **435004**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra radioelektroniky**  
Studijní program: **Komunikace, multimédia a elektronika**  
Studijní obor: **Multimediální technika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Mobilní aplikace pro ovládání laboratorního napájecího zdroje**

Název bakalářské práce anglicky:

**Mobile Application for Control of Laboratory Power Supply**

Pokyny pro vypracování:

Navrhněte a zrealizujte mobilní aplikaci pro platformu Android sloužící k ovládání programovatelného laboratorního napájecího zdroje Manson HCS-3604-USB. Navrhněte a otestujte možnost/možnosti detekce výpadku síťového napájení. Implementujte zejména tyto funkce: nastavení a vyčtení hodnot napětí a proudu, správa profilů (nastavení hodnot napětí a proudu po určitý čas, uložení profilu, načtení profilu), výpočet elektrického náboje a přepnutí mezi profily v případě výpadku síťového napájení. Otestujte a zdokumentujte navrženou aplikaci.

Seznam doporučené literatury:

[1] HCS-3604-USB. Manson [online]. 2013 [cit. 2017-02-02]. Dostupné z: <http://www.manson.com.hk/products/detail/162>  
[2] Android developers. Android [online]. 2017 [cit. 2017-02-02]. Dostupné z: <http://developer.android.com/index.html>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Marek Neruda Ph.D., katedra telekomunikační techniky FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **17.02.2017**

Termín odevzdání bakalářské práce: **26.05.2017**

Platnost zadání bakalářské práce: **30.09.2018**

Podpis vedoucí(ho) práce

Podpis vedoucí(ho) ústavu/katedry

Podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studentky

## Abstrakt

Bakalářská práce se zabývá vývojem mobilní aplikace pro zařízení s operačním systémem Android sloužící k ovládní programovatelného laboratorního zdroje Manson HCS-3604-USB. Teoretická část obsahuje popis zdroje a oficiální aplikace pro jeho ovládní a historii operačního systému Android společně se základy pro vývoj na této platformě. Praktická část je věnována popisu vytvořené aplikace, možnostem napájení mobilního zařízení při komunikaci se zdrojem a testování aplikace.

**Klíčová slova:** Android, Java, mobilní aplikace, napájecí zdroj

**Vedoucí práce:** Ing. Bc. Marek Neruda, Ph.D.

## Abstract

This bachelor thesis describes the development of the Android mobile application for the control of the programmable laboratory power supply Manson HCS-3604-USB. The theoretical part contains the characterization of the power supply and the official application for its control and the history of the Android operating system together with the basics of development on this platform. The practical part is dedicated to the description of the created application, options of charging the mobile device along with communication with the power supply and to the testing of the application.

**Keywords:** Android, Java, mobile application, power supply

**Title translation:** Mobile application for control of laboratory power supply

## Obsah

### 1 Úvod 1

#### Část I Teoretická část

### 2 Napájecí zdroj Manson HCS-3604-USB 4

2.1 Popis..... 4

2.2 Režimy CV a CC ..... 5

2.3 Režimy ovládání..... 5

2.3.1 Normal (Normální)..... 6

2.3.2 Preset (Přednastavené hodnoty) 6

2.3.3 Set (Nastavení) ..... 6

2.3.4 Remote Control (Analogové vzdálené ovládání) ..... 6

2.4 Autotest zdroje..... 6

2.5 Komunikace se zdrojem..... 7

2.6 Aplikace Manson Power Supply remote ..... 7

2.6.1 Popis ..... 7

2.6.2 Funkce ..... 9

2.6.3 Komunikace se zdrojem ..... 10

### 3 Operační systém Android 11

3.1 Historie ..... 11

3.2 Verze OS Android ..... 11

3.2.1 Největší přínosy jednotlivých verzí ..... 12

3.3 Základní komponenty aplikace.. 13

3.3.1 Aktivita (Activity) ..... 13

3.3.2 Služba (Service) ..... 14

3.3.3 Broadcast receiver ..... 15

3.3.4 Poskytovatel obsahu (Content provider) ..... 15

3.4 Manifest ..... 15

3.5 Gradle ..... 15

3.6 Fragment ..... 16

3.6.1 Životní cyklus fragmentu.... 17

3.7 Záměr (Intent) ..... 18

3.8 Podpora různých velikostí obrazovek . . . . .	18	4.1 Ikona . . . . .	27
3.8.1 Density-independent pixel (dip, dp) . . . . .	18	4.2 Vymezení pojmů program a profil v kontextu vytvořené aplikace . . . . .	28
3.8.2 Kvalifikátory denzity (hustoty) obrazových bodů obrazovky . . . . .	18	4.3 Struktura . . . . .	28
3.8.3 Konfigurační kvalifikátory . . . . .	19	4.3.1 MainFragment . . . . .	28
3.8.4 Deklarace velikosti obrazovky v manifestu . . . . .	20	4.3.2 ProfilesFragment . . . . .	30
3.9 Zdroje (Resources) . . . . .	20	4.3.3 AddFragment . . . . .	31
3.10 Uživatelské rozhraní . . . . .	20	4.3.4 UpperLimitFragment . . . . .	32
3.10.1 Komponenty UI . . . . .	20	4.3.5 ProgramListFragment . . . . .	32
3.10.2 Layout . . . . .	21	4.3.6 AddProgramFragment . . . . .	33
3.10.3 Lišty . . . . .	22	4.3.7 ProgramDetailFragment . . . . .	34
3.11 Material Design . . . . .	22	4.3.8 GraphFragment . . . . .	36
3.11.1 Principy . . . . .	22	4.3.9 StatusListFragment . . . . .	37
3.11.2 Popis některých komponent . . . . .	23	4.3.10 SettingsFragment . . . . .	38
<b>Část II</b>		4.4 Funkce . . . . .	38
<b>Praktická část</b>		4.4.1 Komunikace se zdrojem . . . . .	39
<b>4 Vlastní aplikace</b>	<b>27</b>	4.4.2 Ukládání do souboru . . . . .	40
		4.4.3 Posílání zpráv na e-mail . . . . .	40

4.4.4 Detekce výpadku síťového napájení . . . . .	41
4.5 Porovnání s oficiální aplikací . . .	41
4.5.1 UI a podpora zařízení . . . . .	41
4.5.2 Komunikace a napájení . . . . .	42
4.5.3 Vytvoření programů a profilů	42
4.5.4 Grafy . . . . .	43
4.5.5 Posílání e-mailů . . . . .	43
4.5.6 Další . . . . .	43
<b>5 Napájení a komunikace</b>	<b>44</b>
5.1 USB OTG . . . . .	44
5.2 Bezdrátové nabíjení . . . . .	46
<b>6 Testování</b>	<b>47</b>
6.0.1 Testování aplikace . . . . .	47
6.0.2 Testování bezdrátové nabíječky	48
<b>7 Závěr</b>	<b>50</b>
<b>Literatura</b>	<b>51</b>
7.1 Seznam zkratk . . . . .	57

**Přílohy**

**A Tabulky**

**60**

## Obrázky

2.1 Popis zdroje Manson HCS-3604-USB [1] .....	4	4.4 ProfilesFragment s příkladem nastavení profilu na zdroj .....	31
2.2 Uživatelské rozhraní aplikace Manson Power Supply remote, záložka Device [3] .....	8	4.5 AddFragment .....	31
2.3 Tabulka pro vytvoření nového programu v aplikaci Manson Power Supply remote [3] .....	9	4.6 UpperLimitFragment při orientaci zařízení na šířku .....	32
3.1 Schéma životního cyklu aktivity [17] .....	14	4.7 Layout ProgramListFragmentu .	32
3.2 Ukázka použití fragmentů [20] ..	16	4.8 AddProgramFragment spolu se zadááním doby trvání na různých obrazovkách .....	33
3.3 Schéma životního cyklu fragmentu [20] .....	17	4.9 ProgramDetailFragment na různých konfiguracích obrazovky ..	35
3.4 Příklad prvků UI (1. sloupec, shora: Button, EditText, SeekBar; 2. sloupec, shora: CheckBox a RadioButton, Switch) [25] .....	21	4.10 Příklad uloženého grafu (zeleně proud [A], modře napětí [V] a oranžově elektrický náboj [Ah]) ...	37
3.5 Příklad UI dle principů Material designu se zdvihem jednotlivých komponent [37] .....	23	4.11 Příklad StatusListFragmentu ..	37
4.1 Ikona vlastní aplikace .....	27	4.12 Layout SettingsFragmentu ....	38
4.2 MainFragment na různých konfiguracích obrazovky .....	29	6.1 Graf závislosti teploty baterie na čase nabíjení .....	49
4.3 Navigation drawer .....	30	6.2 Graf závislosti úrovně nabití baterie na čase nabíjení .....	49



## Tabulky

6.1 Tabulka testovaných zařízení ... 47

A.1 Příkazy pro komunikaci se  
zdrojem [1] ..... 60

A.2 Přehled verzí OS Android [9] [10] 63

# Kapitola 1

## Úvod

Cílem této bakalářské práce je návrh, realizace a testování mobilní aplikace pro platformu Android sloužící k ovládnání programovatelného laboratorního napájecího zdroje Manson HCS-3604-USB.

Aplikace pro tablety se systémem Android, která umožňuje vzdálenou komunikaci s tímto zdrojem, již existuje, pro komunikaci se zdrojem ovšem potřebuje Raspberry Pi s nainstalovaným driverem, a postrádá některé funkce, především detekci výpadku síťového napájení. Účelem aplikace, která je předmětem mé bakalářské práce, je tedy poskytnout uživateli možnost komunikovat se zdrojem bez nutnosti dalšího zařízení, poskytnout další funkce a zároveň uživateli umožnit komunikovat se zdrojem nejen prostřednictvím tabletu, ale i pomocí mobilního telefonu.

Klíčovými funkcemi aplikace je nastavení a vyčtení hodnoty napětí a proudu a vytvoření, uložení a načtení profilu. Důležitou funkcí je také vytvoření programu včetně jeho uložení a načtení a výpočet elektrického náboje po dobu trvání spuštěného programu. Při detekci výpadku síťového napájení je pak klíčové, aby aplikace umožňovala nastavit na zdroji hodnoty dle uživatelem vybraného profilu.

Tato bakalářská práce je rozdělena na část teoretickou a praktickou. V první kapitole teoretické části je popsán napájecí zdroj Manson HCS-3604-USB spolu s popisem jednotlivých částí a funkcí oficiální aplikace. Druhá kapitola se zabývá operačním systémem Android. Jsou v ní popsány přínosy jednotlivých verzí, komponenty sloužící pro vývoj aplikace a základní principy vytváření uživatelského rozhraní.

Praktická část je věnována vytvořené aplikaci. Obsahuje detailní popis jednotlivých částí aplikace a jejích funkcí, porovnání vytvořené aplikace s oficiální aplikací, seznámení s USB (Universal Serial Bus) On-The-Go, možnosti napájení mobilního zařízení a také testování vytvořené aplikace.



# Část I

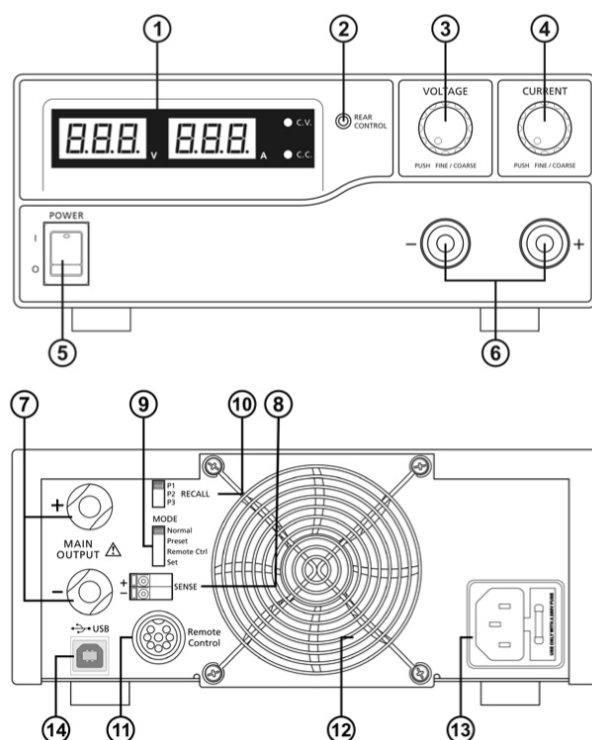
## Teoretická část

## Kapitola 2

### Napájecí zdroj Manson HCS-3604-USB

Jedná se o vzdáleně programovatelný laboratorní napájecí zdroj, který má rozměry (200 x 90 x 275) mm a váží 3,2 kg. Rozsah výstupního napětí je 0,8 V až 60 V a rozsah výstupního proudu činí 0 A až 15 A. Zdroj je možné ovládat pomocí počítačového programu nebo pomocí aplikace přes iPad či tablet s operačním systémem Android [1].

#### 2.1 Popis



Obrázek 2.1: Popis zdroje Manson HCS-3604-USB [1]

1. *"LED displej zobrazující hodnotu napětí a proudu s indikací režimu CV/CC*
2. *Indikátor zadní kontroly (rozsvítí se při použití režimu Preset, Remote Control, nebo Set)*
3. *Otočný regulátor nastavení výstupního napětí*
4. *Otočný regulátor nastavení výstupního proudu*
5. *Hlavní vypínač On/Off*
6. *Pomocné výstupní svorky*
7. *Výstupní svorky*
8. *Model HCS-3604-USB neobsahuje*
9. *Přepínač režimů (Normal, Preset, Remote Control, Set)*
10. *Přepínač paměti*
11. *Konektor pro analogové vzdálené ovládání*
12. *Přívod vzduchu pro chladičí ventilátor*
13. *Zásuvka AC napájení*
14. *USB port (pro vzdálené ovládání jiným zařízením)" (přeloženo z [1])*

## 2.2 Režimy CV a CC

V režimu konstantního napětí (CV) se zdroj snaží udržet na výstupu nastavenou hodnotu napětí pro nastavený rozsah proudu. Pokud zátěž začne odebírat proud vyšší, než je nastavená hodnota, zdroj se přepne do režimu konstantního proudu (CC). V tomto režimu se zdroj snaží udržet nastavenou hodnotu proudu pro nastavený rozsah napětí.

Zdroj mezi režimy přepíná automaticky v závislosti na změně zátěže. Pokud je například zdroj v režimu CC a poklesne odebíraný proud zátěží pod nastavenou hodnotu proudu, automaticky se přepne do režimu CV a naopak [2].

## 2.3 Režimy ovládání

Z katalogového listu zdroje [1] lze vyčíst, že má celkem 4 režimy ovládání, mezi kterými lze přepínat pomocí přepínače režimů na zadní straně zdroje (9).

### ■ 2.3.1 Normal (Normální)

Tento režim umožňuje nastavit výstupní napětí a proud prostřednictvím dvojčinných regulátorů (3 a 4), které umožňují hrubé a jemné nastavení hodnoty. Pro změnu mezi hrubým a jemným režimem je potřeba regulátor stisknout [1].

Režim je také nastaven při vzdáleném ovládní zdroje pomocí některého z dostupných softwarů.

### ■ 2.3.2 Preset (Přednastavené hodnoty)

*"V tomto režimu je rozsvícen indikátor zadní kontroly (2), čímž je signalizována deaktivace regulátorů napětí a proudu"* [1]. Režim nastavuje na zdroji hodnoty na základě hodnot uložených ve vybrané paměti, kterou lze zvolit přepínačem paměti (10). Na zdroji mohou být takto uloženy zároveň 3 hodnoty napětí a proudu, přičemž každá paměť má tyto hodnoty továrně přednastaveny a uživatel je může nahradit svými (viz následující bod) [1].

### ■ 2.3.3 Set (Nastavení)

Tento režim slouží k nastavení hodnot napětí a proudu ve 3 pamětech zdroje. Hodnoty v paměti se nastaví zvolením jedné ze tří pamětí pomocí přepínače paměti (10) a následným nastavením hodnot prostřednictvím regulátorů (3 a 4). Pro potvrzení nastavení je nutné vrátit pozici přepínače režimů (9) z režimu Set do režimu Preset. Nastavené hodnoty budou v paměti uchovány i po vypnutí zdroje [1].

### ■ 2.3.4 Remote Control (Analogové vzdálené ovládní)

Režim Remote Control slouží *"pro ovládní výstupního napětí a proudu pomocí konektoru pro analogové vzdálené ovládní"* [1].

## ■ 2.4 Autotest zdroje

Při každém spuštění zdroj provádí sekvenci testů pro kontrolu svého stavu. Bezprostředně po spuštění se na displeji zobrazí verze softwaru, následuje zkouška segmentů displeje, testy světelných indikátorů, kontrola ochrany proti přepětí, proti přetížení a proti přehřátí a nakonec proběhne kontrola ventilátoru a vypnutí výstupu [1].

## 2.5 Komunikace se zdrojem

Komunikace se zdrojem probíhá přes COM port pomocí ASCII příkazů. Seznam všech příkazů, včetně jejich popisů a příkladů, je uveden v tabulce A.1 v příloze.

"Výchozí nastavení COM portu:

- Přenosová rychlost (*baud-rate*): 9600
- Počet datových bitů (*data bits*): 8
- Počet stop bitů (*stop bits*): 1
- Parita (*parity*): žádná (*None*) (přeloženo z [1])

## 2.6 Aplikace Manson Power Supply remote

Pro vzdálenou komunikaci se zdrojem HCS-3604-USB, případně dalšími produkty firmy Manson, vyvinula firma oficiální aplikaci jak pro iPady, tak i pro tablety se systémem Android. Aplikace pro platformu Android je dostupná z internetového obchodu Google Play. Je možné ji nainstalovat na zařízení s verzí Android 4.0 a vyšší a je určena pro tablety, které mají oproti mobilním telefonům větší rozměry obrazovky [3].

### 2.6.1 Popis

Aplikace má své uživatelské rozhraní rozděleno do pěti částí – Device, Monitor, Program, Information a Settings.

#### Device

V této části je možné přidat či upravit seznam používaných zařízení a zároveň se s vybraným zařízením spojit a s daným zařízením komunikovat.

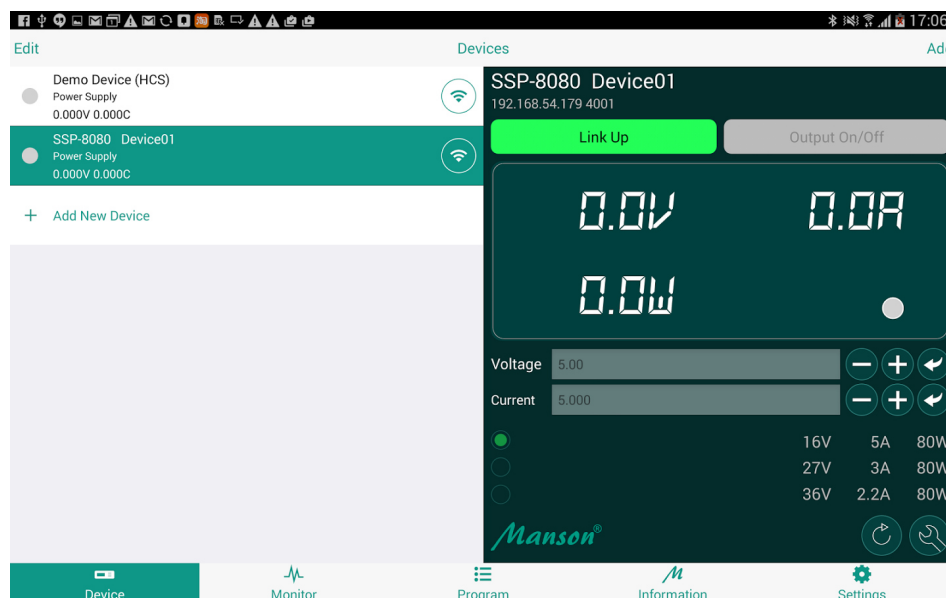
UI této části je možné vidět na obrázku 2.2. V levé části obrazovky se nachází seznam zařízení spolu s demoverzí zařízení, které uživateli usnadňuje seznámení se s funkcemi aplikace bez nutnosti být připojen k reálnému zařízení.

Přidání nového zařízení začíná výběrem konkrétního zařízení ze seznamu zařízení podporovaných aplikací (v mém případě HCS-3604-USB). Dále se zobrazí formulář, do kterého uživatel vyplní název, volitelnou poznámku a nastavení komunikace, do kterého se zadává IP adresa Raspberry Pi a číslo portu, ke kterému je zařízení připojeno.

Pravá polovina záložky *Device* je věnována komunikaci se zařízením. Umožňuje uživateli připojit se ke zdroji, či se od něj odpojit, zapnout či vypnout výstup zdroje a jednotlivě nastavit napětí a proud na zdroji.



Zobrazuje aktuální hodnoty napětí, proudu a výkonu na zdroji, režim, ve kterém pracuje a zda je zapnut, či vypnut výstup zdroje. Také se zde zobrazuje horní limit napětí a proudu a maximální výkon.



**Obrázek 2.2:** Uživatelské rozhraní aplikace Manson Power Supply remote, záložka Device [3]

## Monitor

Tato část umožňuje sledovat změny hodnot napětí a proudu v reálném čase. Lze volit mezi zobrazením dat v grafu či tabulce. Levá polovina obrazovky obsahuje seznam monitorovacích souborů. Je možné přidat nové soubory a staré mazat.

Na pravé polovině se pak zobrazuje detail vybraného souboru s možností jej pojmenovat a vybrat zařízení, které bude monitorováno. Pro spuštění sledování je nutné, aby toto zařízení bylo připojeno k tabletu v záložce *Device*.

Uživatel má možnost nastavit interval, ve kterém budou hodnoty ze zdroje čteny. Je možno nastavit 1 sekundu, ale i 5 hodin. Výsledná data je možné exportovat a graf je možné uložit jako obrázek. Graf je možné dvojitým poklepáním zvětšit přes celou obrazovku zařízení, nelze nicméně část grafu přiblížit.

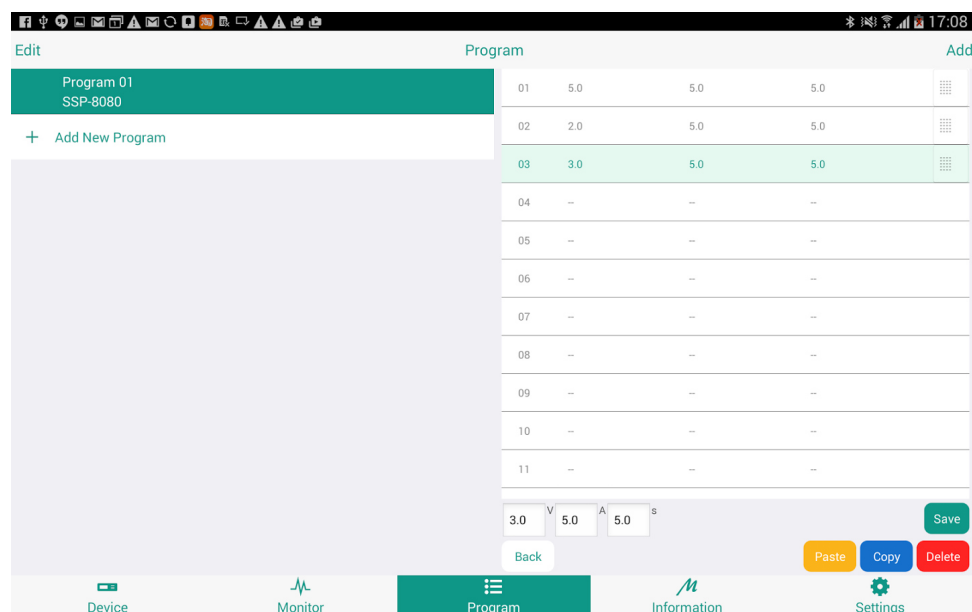
## Program

Záložka *Program*, kterou lze vidět na obrázku 2.3, umožňuje posílat na vybrané zařízení uživatelem vytvořenou sekvenci hodnot napětí a proudu, přičemž jednotlivé hodnoty se posílají po uživatelem zadaném čase.

Uživatel vyplní tabulku, do jejíž řádků zadá postupně hodnoty napětí

a proudu, které se na zdroji mají nastavit, a čas v sekundách, po který se má dané nastavení na zdroji udržet. Pořadí jednotlivých řádků lze měnit, počet řádků, tedy změn nastavených hodnot v čase, je ovšem omezen na 20. Takto vytvořenou posloupnost lze spustit v až 999 cyklech.

Po nastavení hodnot a dob jejich trvání se pak vytvořený program zobrazí v grafu, kde na vertikální ose jsou hodnoty nastaveného napětí a proudu a na horizontální ose je čas. Nad grafem se v textové podobě zobrazuje hodnota napětí a proudu, která je na zdroji momentálně nastavena, číslo cyklu, který se právě provádí, a čas, po jaký program běží.



**Obrázek 2.3:** Tabulka pro vytvoření nového programu v aplikaci Manson Power Supply remote [3]

Uživatelé vytvořenou tabulku aplikace zobrazí v přehledném grafu, ve kterém je hodnota nastaveného napětí na zařízení v závislosti na čase vykreslena modře a hodnota nastaveného proudu červeně.

## Information a Settings

Část *Information* obsahuje informaci o aplikaci a firmě Manson a externí odkazy související s aplikací.

V záložce *Settings* je pak možné upravit nastavení pro spojení se zařízením.

### 2.6.2 Funkce

Aplikace má tyto funkce:

- nastavení hodnoty napětí a proudu
- zapnutí/vypnutí výstupu zdroje

- virtuální displej zdroje (aktuální hodnoty napětí, proudu a výkonu, režim CC/CV a indikace zapnutého/vypnutého výstupu)
- zobrazení horního limitu napětí a proudu a hodnoty maximálního výkonu
- sledování a zobrazení hodnot napětí a proudu v reálném čase v textové či grafické podobě s možností měnit interval sledování (1 sekunda až 5 hodin) a uložit či exportovat výsledek monitorování
- vytvoření programu (omezeno jen na 20 změn)
- opakování programu ve více cyklech (1 až 999 cyklů)

### 2.6.3 Komunikace se zdrojem

Aplikace pro komunikaci se zařízením potřebuje Raspberry Pi, na kterém musí být nainstalován potřebný driver, který lze stáhnout z oficiálních webových stránek firmy Manson[4] a který slouží k připojení zařízení k Raspberry Pi přes USB port. Následně je možné zdroj ovládat pomocí oficiální aplikace, nainstalované na tabletu, přes protokol TCP/IP [5].

## Kapitola 3

### Operační systém Android

Android je open-source platforma založená na jádře Linuxu a vyvíjená organizací Open Handset Alliance, již tvoří desítky firem. Jde například o společnost Google, HTC, Intel či Samsung. Jako jeden z mála operačních systémů (OS) podporuje Android více platform. Jeho výhodou, ale zároveň i nevýhodou, je jeho otevřenost a možnost úprav, které se mohou týkat i firmwaru. Android nabízí volnost jak vývojářům, tak i uživatelům [8].

Od roku 2011 je Android světově nejrozšířenějším operačním systémem v chytrých telefonech. V posledních letech Android dosahuje přes 80% podílu tržeb na celosvětovém trhu [6]. V České republice se poslední dva roky drží nad 70% podílu tržeb [7]. Dá se předpokládat, že svou převahu si v následujících letech Android ponechá. Jediným skutečným konkurentem je pro Android operační systém iOS, který vyvíjí společnost Apple. Ten na trhu v současné době dosahuje necelých 20%.

#### 3.1 Historie

Společnost Android Inc. vznikla v roce 2003 a o dva roky později ji odkoupil Google za 50 milionů dolarů. O další dva roky později, v roce 2007, byla založena Open Handset Alliance, která za vývojem stojí dodnes. V téže roce vyšel také vývojářský kit (SDK). O rok později přišel v USA na trh první chytrý mobilní telefon s operačním systémem Android 1.0, ten na trhu ovšem velmi úspěšný nebyl. První velký úspěch v tržbách systém zaznamenal v roce 2009 s verzí Android 1.5 (Cupcake) [8].

#### 3.2 Verze OS Android

Každá verze systému Android nese číselné a také kódové označení, kterým je název sladkosti. Názvy sladkostí jsou voleny tak, že počáteční písmena jdou s každou novou verzí za sebou podle abecedy [8].

V tabulce A.2 v příloze lze nalézt seznam všech dosud vydaných verzí. Je zde uvedeno číselné i kódové označení, verze API (Application Programming Interface - rozhraní pro programování aplikací) a měsíc a rok vydání dané verze.

### ■ 3.2.1 Největší přínosy jednotlivých verzí

**Cupcake** - umístování widgetů na domácí obrazovku, nahrávání a přehrávání videa, virtuální klávesnice třetích stran s vlastními slovníky, automatické otáčení obrazovky

**Donut** - vylepšení Android Marketu (předchůdce Google Play), podpora různých velikostí obrazovek a displejů s vyšším rozlišením (WVGA), univerzální vyhledávač

**Eclair** - představení mapové navigace Google, hlasové zadávání, prohlížeč s podporou HTML5, podpora Bluetooth 2.1, rozšířené funkce fotoaparátu, nový design uživatelského prostředí, přidání gesta pro přiblížení a oddálení obrazovky, živé tapety

**Froyo** - hlasové ovládání, možnost vytvořit Wi-Fi hotspot, výrazné zlepšení výkonu procesoru, instalace a přesun aplikací z vnitřní paměti zařízení na SD kartu

**Gingerbread** - podpora NFC (Near Field Communication), vylepšená správa napájení, vylepšené kopírování, přidání "velikonočního vajíčka"- skryté minihry, podpora gyroskopu a barometru

**Honeycomb** - pouze pro tablety, nová systémová lišta, rychlý přístup k základnímu nastavení, podpora USB příslušenství

**Ice Cream Sandwich** - sdílení obsahu přes NFC, navigace na obrazovce, vytváření složek na domovské obrazovce, nová gesta, kontrola využití dat, Full HD videa

**Jelly Bean** - Android Market změněn na Google Play, vylepšené notifikace, podpora více účtů na jednom zařízení, widgety na uzamčené obrazovce, zrychlení vykreslování obrazu

**KitKat** - hlasové ovládání pomocí hlášky "OK Google", možnost roztáhnutí aplikací přes celou obrazovku, zvýšení výkonu, větší zabezpečení NFC plateb

**Lollipop** - notifikace na uzamykací obrazovce, designové principy "Material Design", Android TV, odemknutí zařízení v okolí známého Bluetooth zařízení, podpora RAW fotografií, adaptivní nastavení jasu

[8] [11] [12]

**Marshmallow** - zapínání a vypínání oprávnění aplikací, snížení spotřeby baterie, otisky prstů [13]

**Nougat** - možnost použít více jazyků zároveň, více aplikací na jedné obrazovce zároveň, podpora virtuální reality, možnost přizpůsobení rychlého nastavení, šifrování souborů, rychlejší spuštění zařízení [14]

Dle dostupných dat z května 2017 jsou v současnosti nejpoužívanějšími verzemi Lollipop a Marshmallow. Obě jsou používány na více než

30% zařízeních s OS Android. Poměrně hojně je též využívána verze KitKat, nainstalována je na téměř 20% zařízeních. O něco méně, s podílem necelých 10%, je pak používána verze Jelly Bean. Prozatím nejnovější verzi, Nougat, má nainstalováno přes 7% zařízení s Androidem. Tato verze zároveň zaznamenává nejvyšší nárůst v používání. Starší verze se již téměř nepoužívají [15].

## 3.3 Základní komponenty aplikace

### 3.3.1 Aktivita (Activity)

Aktivita se uživateli zobrazí po spuštění aplikace a slouží pro interakci s uživatelem prostřednictvím implementovaného GUI. Lze si ji představit jako jednu obrazovku s uživatelským prostředím v dané aplikaci. Aplikace může mít aktivit více, přičemž jednotlivé aktivity jsou na sobě vzájemně nezávislé a každá z těchto aktivit může být spuštěna prostřednictvím jiné aplikace. Je například možné z aplikace fotoaparátu spustit jinou aktivitu v aplikaci pro posílání e-mailu a vložit do posílané zprávy pořízenou fotografii [16].

#### Životní cyklus aktivity

*"Aktivity jsou v systému spravovány v zásobníku aktivit. Když se spustí nová aktivita, zařadí se na vrchol zásobníku a stane se běžící aktivitou - předchozí aktivita vždy v zásobníku zůstane pod ní a nepřejde do popředí, dokud nová aktivita neskončí." (přeloženo z [17])*

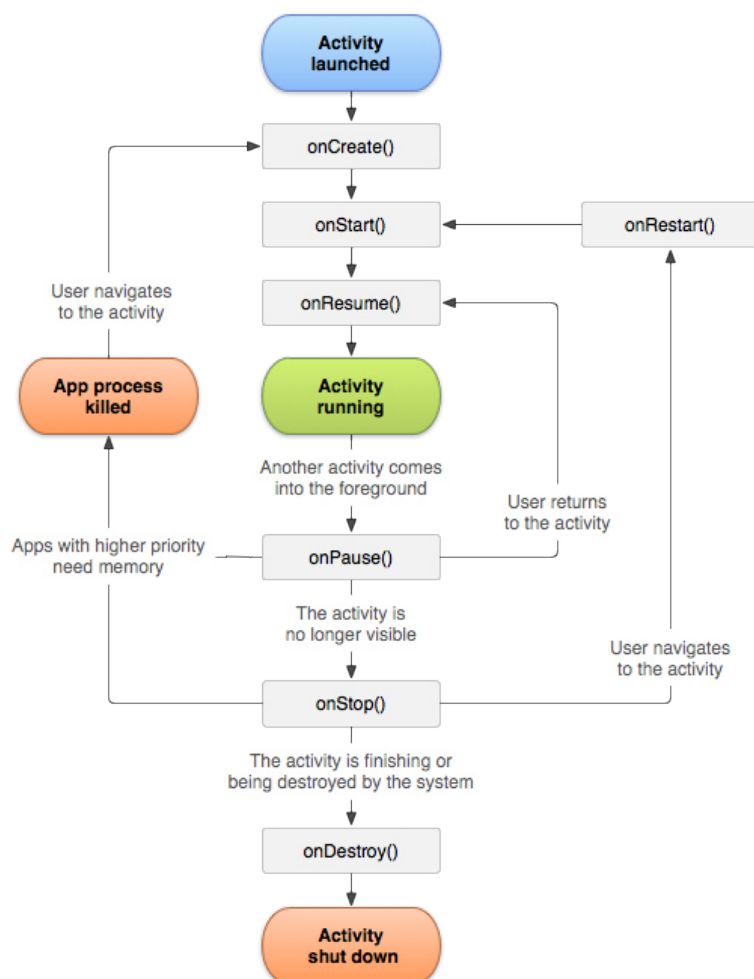
#### Aktivita se může nacházet ve 4 stavech:

- *Aktivní (běžící)* - je na popředí
- *Pozastavená* - aktivita je částečně viditelná, ale uživatel s ní nemůže interagovat, může být ukončena v případě extrémního nedostatku paměti
- *Zastavená* - aktivita je zcela překryta jinou aktivitou, zůstává v paměti
- *Zničená* - aktivita je odstraněna z paměti, při opětovném spuštění musí být zcela restartována [17]

#### Popis jednotlivých metod:

- *onCreate()* - volá se vždy při prvním vytvoření aktivity, vždy následuje metoda *onStart()*
- *onRestart()* - volá se, pokud aktivita byla zastavena, následuje metoda *onStart()*
- *onStart()* - aktivita se stává viditelnou pro uživatele
- *onResume()* - aktivita začíná interagovat s uživatelem, nachází se na vrcholu zásobníku aktivit

- *onPause()* - aktivita je pozastavena, je vhodné uložit změny či zastavit animace a další prvky, které by zbytečně využívaly procesor
- *onStop()* - aktivita je zastavena
- *onDestroy()* - poslední metoda volaná před zničením aktivity, buď je ukončena metodou *finish()*, nebo zničena pro ušetření místa v paměti [17]



Obrázek 3.1: Schéma životního cyklu aktivity [17]

### 3.3.2 Služba (Service)

Služby se používají pro běh aplikace na pozadí při realizaci dlouhotrvajících operací či práci se vzdálenými procesy. Neposkytují uživatelské rozhraní. Pomocí služby je například možné přehrávat hudbu, zatímco uživatel používá jinou aplikaci.

Služba může být vytvořena jinou komponentou a svázána s ní pro vzájemnou interakci. Uživatel si může být jejího běhu vědom, například prostřednictvím notifikace, ale také může probíhat bez jeho vědomí. V takovém případě ji systém může zrušit, pokud bude potřebovat místo v operační paměti [16].

### 3.3.3 Broadcast receiver

Broadcast receiver umožňuje přenášet do aplikace události, které se odehrávají na zařízení, a umožňují tak aplikaci, i když v danou chvíli není spuštěna, reagovat na tyto události. Vhodným příkladem může být oznámení o nízkém stavu baterie, doručení SMS, či zobrazení příchozího hovoru. Nemají uživatelské rozhraní, ale mohou vytvářet notifikace. Přenosy jsou do aplikace doručovány pomocí objektů typu Intent (záměr) [16].

### 3.3.4 Poskytovatel obsahu (Content provider)

Třídou a Content Provider slouží ke správě sdíleného obsahu aplikace. Každá aplikace, která má potřebné oprávnění, může získávat či zaznamenávat data pomocí poskytovatele obsahu [16].

## 3.4 Manifest

Každá komponenta musí být deklarována v manifestu aplikace. Jedná se o soubor AndroidManifest.xml, který je nezbytnou součástí každé aplikace a musí se nacházet v kořenovém adresáři projektu. Obsahuje základní informace, které systém potřebuje před samotným spuštěním aplikace. Mimo jiné je do něj také nutné uvést všechna oprávnění, která aplikace potřebuje pro své fungování, deklarovat v něm minimální úroveň API, kterou aplikace vyžaduje, a také název balíčku Java, který následně pro aplikaci slouží jako unikátní identifikátor [18].

## 3.5 Gradle

Gradle je nástroj pro sestavení (build) programu, který vývojové prostředí Android Studio používá pro automatizaci a správu sestavovacího procesu. Android Studio se při založení nového projektu samo postará o vytvoření několika souborů Gradle a také o nastavení základní konfigurace. Jedná se například o soubory settings.gradle, který specifikuje zahrnuté moduly, a build.gradle, který podle toho, na jaké úrovni se nachází, definuje konfigurace buď pro všechny moduly v projektu, či pouze pro konkrétní modul. Většina aplikací má pouze jeden modul.

Pokud má aplikace více verzí, například bezplatnou a placenou, je možné manuálně v souboru Gradle specifikovat použité kódy a zdroje pro danou verzi. Nejčastějším zásahem do toho souboru ale bývá přidání dependence na použité knihovny. Není tak nutné knihovnu manuálně stahovat a kopírovat do složky projektu.

Po každé změně souboru Gradle je nutné jej synchronizovat s projektem [19].



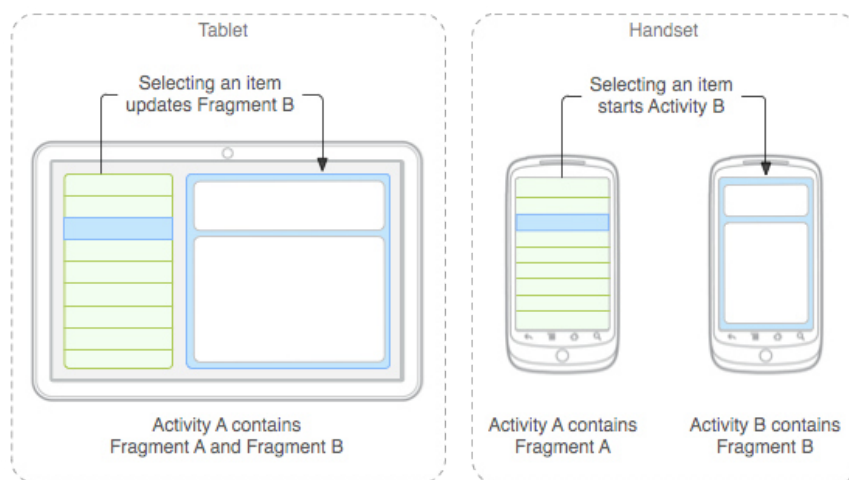
## 3.6 Fragment

Fragmenty jsou komponenty, které byly představeny ve verzi 3.0 (Honeycomb) a jsou součástí aktivity. Představují část jejího UI (User Interface - uživatelské rozhraní) či chování. Výhodou fragmentů je, že je možné je použít ve více aktivitách a také to, že v jedné aktivitě může být fragmentů umístěno více, přičemž každý z fragmentů může být individuálně přidáván a odebírán za běhu aktivity.

Změnu fragmentu, která při nějaké operaci nastane, je možné uchovat v zásobníku (back stack), který uživateli umožní tuto operaci zvrátit stisknutím navigačního tlačítka Zpět. Na každou operaci je také možné aplikovat animaci.

Fragment slouží jako náhrada aktivity, kterou je možné použít vícekrát, a má svůj vlastní životní cyklus, který je přímo ovlivněn životním cyklem aktivity, ve které je vložený. Jeho hlavním účelem je podpora dynamického a flexibilního návrhu UI, především na větších obrazovkách tabletů.

Dobrým příkladem využití fragmentů je případ, který je znázorněn na obrázku 3.2, kde aplikace obsahuje jiné konfigurace layoutu pro tablety a mobilní telefony. Díky velké velikosti obrazovky tabletu je vhodné zde zobrazit dva fragmenty, například fragment se seznamem položek a fragment s náhledem položky, vedle sebe v jedné aktivitě. Na menší obrazovce mobilního telefonu je nicméně vhodnější zobrazit seznam položek v jedné aktivitě a náhled položky v další aktivitě.



**Obrázek 3.2:** Ukázka použití fragmentů [20]

Většina fragmentů má své vlastní UI, které přispívá k vlastnímu layoutu aktivity, ve které jsou umístěny. Není to ale podmínkou. Je možné vytvořit fragment k poskytnutí procesu na pozadí, který UI nepotřebuje.

Fragmenty jsou v aktivitě spravovány pomocí třídy `FragmentManager`, díky které je například možné získat fragmenty uvnitř aktivity.

Komunikace fragmentu s aktivitou je možná definováním callbacku (zpětného volání) uvnitř fragmentu a jeho implementováním v hostitelské aktivitě. Když aktivita přes rozhraní obdrží callback, může přijaté informace sdílet také s ostatními fragmenty v layoutu [20].

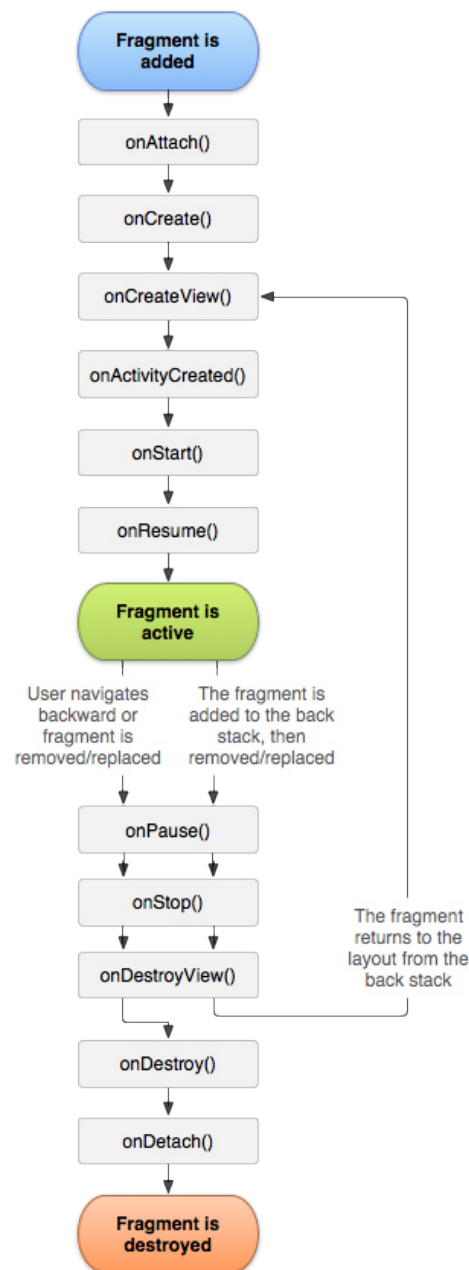
### 3.6.1 Životní cyklus fragmentu

Životní cyklus fragmentu je podobný životnímu cyklu aktivity a je s ním přímo spjatý. Pokud je například aktivita pozastavena voláním metody `onPause()`, obdrží toto volání i každý fragment uvnitř této aktivity a je pozastaven i daný fragment.

Největším rozdílem je to, že fragment je umístěn do zásobníku jen tehdy, pokud o to vývojář výslovně požádá voláním metody `addToBackStack()`.

Životní cyklus fragmentu kromě metod společných s aktivitou (`onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()` a `onDestroy()`) sestává z pěti dalších metod:

- `onAttach()`  
- fragment je přidružen k aktivitě
- `onCreateView()`  
- vytvoří se UI fragmentu
- `onActivityCreated()`  
- volá se po dokončení metody `onCreate()` v hostitelské aktivitě
- `onDestroyView()`  
- je odstraněno UI fragmentu
- `onDetach()`  
- fragment je odloučen od aktivity [20]



**Obrázek 3.3:** Schéma životního cyklu fragmentu [20]

## 3.7 Záměr (Intent)

*"Intent je asynchronní zpráva, která nese informaci o požadované akci. Akcí může být spuštění aktivity, služby, případně uživatelsky definovaná akce" [8].*

Je možné vytvořit explicitní záměr k vytvoření akce použitím jména komponenty (názvu třídy), kterou systém ihned spustí (např. službu pro stáhnutí souboru na pozadí), nebo implicitním záměrem deklarovat obecnou akci, která má nastat (např. otevření jiné aplikace k zobrazení polohy na mapě) - systém najde příslušnou komponentu porovnáním záměru s filtry záměrů (intent filters), které jsou deklarovány v manifestu všech aplikací nainstalovaných na zařízení. Pokud je záměr kompatibilní s více aplikacemi, zobrazí se uživateli dialogové okno, které mu umožní si aplikaci vybrat [21].

## 3.8 Podpora různých velikostí obrazovek

Jednou z výhod OS Android je, že od verze 1.6 (Donut) podporuje zařízení s různou velikostí a rozlišením obrazovky. Je ovšem žádoucí, aby si uživatel každého zařízení připadal, jako by byla aplikace vytvořena přímo pro jeho zařízení.

Většinu práce pro optimalizaci vzhledu pro různá zařízení vykonává systém Android automaticky změnami měřítka a velikosti prvků. I přesto je vhodné, aby vzhled na různých konfiguracích obrazovky přizpůsobil sám vývojář poskytnutím alternativního layoutu, čímž se zážitek uživatele aplikace znásobí.

Je žádoucí, aby na všech typech obrazovek byly vidět všechny prvky UI a aby layout na obrazovkách efektivně využíval dostupného místa [22].

### 3.8.1 Density-independent pixel (dip, dp)

Jedná se o virtuální jednotku pixelu, která se používá při návrhu UI. Systém v závislosti na hustotě obrazových bodů obrazovky mění měřítko dp tak, že z pohledu uživatele je zachována fyzická velikost elementů UI při zobrazení na obrazovkách s různým DPI. Jeden dp je ekvivalentem k jednomu pixelu na 160 DPI (dots per inch) obrazovce, takže například na obrazovce s 240 DPI se 1 dp rovná 1,5 fyzického pixelu [22].

### 3.8.2 Kvalifikátory denzity (hustoty) obrazových bodů obrazovky

Rozdělují se do šesti skupin:

- *ldpi (nízká) 120dpi*
- *mdpi (střední) 160dpi*
- *hdpi (vysoká) 240dpi*

- *xhdpi (velmi vysoká) 320dpi*
- *xxhdpi (velmi velmi vysoká) 480dpi*
- *xxxhdpi (velmi velmi velmi vysoká) 640dpi"*

(přeloženo z [22])

Především u obrázků je vhodné vytvořit alternativní verze v různých rozlišeních pro různé denzity obrazovek, jelikož změna měřítka bitmapy může způsobit artefakty. Pokud alternativní verze pro danou denzitu v aplikaci neexistuje, systém buď použije výchozí verzi (bez kvalifikátoru) a změní jeho měřítko s ohledem na konkrétní obrazovku, nebo místo výchozí verze použije verzi s kvalifikátorem, jejíž změna měřítka způsobí méně artefaktů. Na obrázky, u nichž nemá docházet ke změně měřítka, je nutné použít kvalifikátor "nodpi"[22].

### ■ Kvalifikátory orientace

Kvalifikátory na základě orientace obrazovky z pohledu uživatele - na šířku (land), nebo na výšku (port) [22].

### ■ Kvalifikátory velikosti

Dříve se velikost obrazovek rozdělovala podle délky změřené na diagonále a členila se do 4 skupin. Od verze 3.2 se toto dělení nepoužívá a nahradilo jej dělení podle nejkratší šířky, dostupné šířky a dostupné výšky obrazovky:

- *Nejkratší šířka obrazovky* - Kvalifikátor má formát `sw<N>dp` (např. `sw600dp`) a zajistí, že má aplikace k dispozici alespoň `<N> dp` šířky pro její UI při jakékoliv orientaci obrazovky (hodnota se při otočení obrazovky nemění a započítává se do ní jen prostor, který je opravdu dostupný pro UI aplikace - může být tedy zmenšen například o status bar). Nejkratší šířka je často rozhodujícím faktorem, jelikož ve většině aplikací se roluje ve vertikálním směru a vodorovný prostor je omezen.
- *Dostupná šířka obrazovky* - Kvalifikátor má formát `w<N>dp` (např. `w720dp`) a jeho hodnota se mění se změnou orientace zařízení.
- *Dostupná výška obrazovky* - Kvalifikátor má formát `h<N>dp` (např. `h720dp`) a je podobný předchozímu, jen je založen na dostupné výšce, nikoliv šířce. Ve většině aplikací není nutný, neboť v aplikacích často bývá rolováno ve vertikálním směru [22].

### ■ 3.8.3 Konfigurační kvalifikátory

Nová konfigurace obrazovky se vytvoří přidáním nové složky v adresáři se zdroji ve formátu `<název_zdroje>-<kvalifikátor>`. Pro jednu konfiguraci je možné použít více kvalifikátorů zároveň (např. `res/layout/sw600dp-land`) [22].

### ■ 3.8.4 Deklarace velikosti obrazovky v manifestu

Velikost podporovaných obrazovek je možné deklarovat explicitně v manifestu. Tím se zajistí, že nebude možné stáhnout aplikaci na zařízení, jejichž velikost obrazovky není podporována.

Například pomocí atributu "android:requiresSmallestWidthDp" je specifikována minimální potřebná nejmenší šířka obrazovky v jednotkách dp, která musí být k dispozici pro UI. Jelikož minimální velikost obrazovky podporované Androidem je 426dp x 320dp, není nutné jej použít, pokud aplikace podporuje všechny velikosti obrazovek počínajíc touto [22].

## ■ 3.9 Zdroje (Resources)

Zdroje se v projektu vkládají do složky *res*, která má několik podsložek rozdělujících zdroje podle jejich typu:

- *animator* a *anim* - XML soubory definující animace (rozděleny na základní a pokročilejší animace)
- *color* - XML soubory definující barvy pro různé stavy objektů, na které jsou aplikovány
- *drawable* - bitmapové soubory (.png, .9.png, .jpg, .gif) či XML soubory, které mohou definovat například tvary
- *mipmap* - obsahuje ikonu aplikace v různém rozlišení
- *layout* - XML soubory definující layout UI
- *menu* - XML soubory definující menu
- *raw* - libovolné soubory uložené ve své syrové podobě
- *values* - XML soubory definující například použité řetězce znaků, barvy či styly
- *xml* - libovolné XML soubory

Všechny zdroje jsou přístupné pomocí jejich ID, která jsou definována ve třídě R projektu [23].

## ■ 3.10 Uživatelské rozhraní

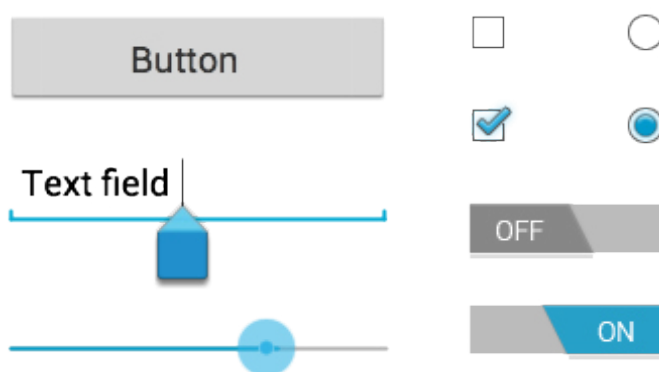
*"Všechny prvky UI v aplikaci pro Android jsou vytvořeny pomocí objektů View a ViewGroup. View je objekt, který na obrazovce vykresluje něco, s čím uživatel může interagovat. ViewGroup je objekt, který obsahuje jiné objekty View (a ViewGroup) pro definování layoutu UI."* (přeloženo z [24])

### ■ 3.10.1 Komponenty UI

Nejčastěji využívané prvky UI:

- *TextView* - textové pole

- *EditText* - editační okno, které očekává textový vstup od uživatele
- *ImageView* - zobrazení obrázku
- *Button* - tlačítko
- *ImageButton* - tlačítko, na kterém je místo textu zobrazen obrázek
- *Switch* - přepínač mezi dvěma stavy
- *CheckBox* - zaškrtačací pole
- *RadioButton* - výběr jedné možnosti z více možností - používá se několik těchto prvků najednou
- *ProgressBar* - indikace probíhající činnosti
- *SeekBar* - rozšíření předchozího prvku s přidáním posuvníkem, který uživateli umožňuje nastavit hodnotu [8]



**Obrázek 3.4:** Příklad prvků UI (1. sloupec, shora: Button, EditText, SeekBar; 2. sloupec, shora: CheckBox a RadioButton, Switch) [25]

### 3.10.2 Layout

Nejběžnější typy layoutů:

- *LinearLayout* - vnořené prvky uspořádány buď v jednom sloupci pod sebou, nebo v jednom řádku vedle sebe
- *RelativeLayout* - layout je tvořen na základě vzájemných pozic jednotlivých prvků, ale také vzájemné pozice k rodičovskému layoutu
- *FrameLayout* - tento layout je navržen pro zobrazení jediného objektu View a využívá se především u dynamicky vytvářených fragmentů na starších verzích Androidu a pro zobrazení složitějších prvků
- *TableLayout* - layout využívaný do verze 4.0 (Ice Cream Sandwich) pro tvorbu tabulek, nahradil ho nový kontejner GridLayout
- *GridLayout* - layout ve formě matice prvků [8]

FrameLayout má několik podtříd. Za zmínku stojí DatePicker, který umožňuje vybrat datum [26], NavigationView, který zobrazuje menu v Navigation draweru (postranní navigaci) [27], a především ScrollView, který uživateli umožňuje rolovat ve vertikálním směru (pro horizontální rolování se používá kontejner HorizontalScrollView) [28].

Pro dynamický obsah je možné použít layout, který je podtřídou třídy `AdapterView`. V té jsou data k layoutu vázána pomocí adaptéru, díky čemuž je možné měnit obsah za běhu aplikace [29]. Mezi takovéto layouty patří:

- *GridView* - zobrazení prvků ve dvourozměrné mřížce, ve které je možné rolovat [30]
- *ListView* - vytvoření seznamu s položkami, který je možné rolovat [31]
- *Spinner* - umožňuje výběr položky z rozevírací nabídky [32]

### 3.10.3 Lišty

Součástí téměř všech aplikací na Android je app bar, dříve známý též jako action bar, který poskytuje navigaci v aplikaci a také akce s danou aplikací spojené. Standardně obsahuje název aplikace či konkrétní stránky, ikonu navigace a akční ikony spolu s ikonou menu. Je možné do této lišty přidat i vyhledávání.

Kromě app baru existují také systémové lišty, které se dělí na status bar a navigation bar. Status bar je na obrazovce umístěný nahoře a obsahuje notifikační a systémové ikony, například aktuální čas či stav baterie. Navigation bar se naopak zobrazuje na spodní části obrazovky a zobrazuje navigační tlačítka zařízení - zpět, domů a přehled [33].

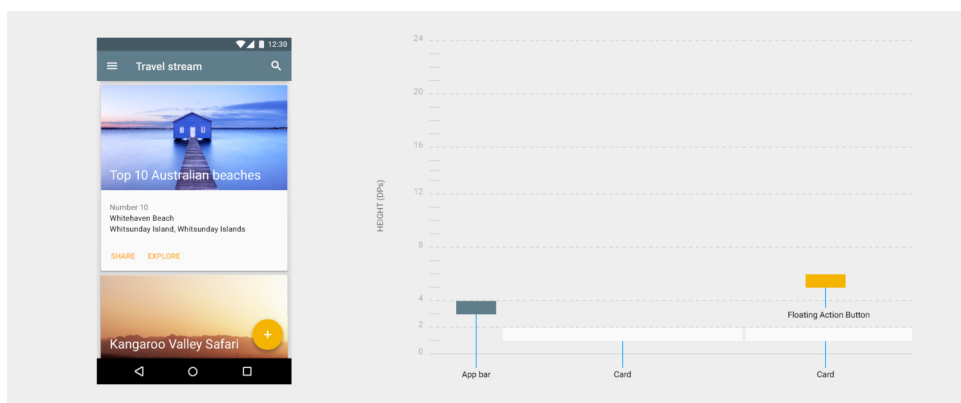
## 3.11 Material Design

Verze Android 5.0 (Lollipop) představila nové principy designu zvané Material design. Pro starší verze je možné použít knihovnu Android Design Support Library, která je kompatibilní s verzí Android 2.3 (Gingerbread) a vyšší. Tak je možné i ve starších verzích použít komponenty využívané Material designem [34].

### 3.11.1 Principy

Material design vychází z hmatatelné reality a je inspirován listem papíru umístěným do trojrozměrného prostředí, přičemž osa  $z$  je umístěna kolmo na rovinu displeje a prvky umístěné na nejvyšším bodě této osy jsou uživatelům zobrazeny jako nejbližší. Každý list materiálu má přitom standardně tloušťku 1dp [35], přičemž obsah zobrazený na něm tloušťku nepřidává [36].

Celé prostředí je osvětleno dvěma zdroji světla, které vytvářejí tvrdší směrové stíny a jemnější stíny ve všech směrech kolem listu materiálu. Stíny jsou závislé na pozici jednotlivých listů v ose  $z$  [35]. List materiálu nelze ohýbat, ale může měnit svůj tvar a velikost v rovině (v osách  $x$  a  $y$ ). Materiály je možné slučovat a oddělovat a posouvat po libovolné ose, ale v jednom bodě prostoru nesmí být zobrazeno více materiálů zároveň a při změně zdvihu (polohy na ose  $z$ ) nesmí procházet přes sebe [36].



**Obrázek 3.5:** Příklad UI dle principů Material designu se zdvihem jednotlivých komponent [37]

Material design přišel i s barevnou paletou, která obsahuje barvy hlavní (v aplikaci se vyskytují nejčastěji) a doplňkové (zdůrazňují klíčové části aplikace). V aplikaci je možné použít i světlejší a tmavší verze těchto barev, vždy je ale potřeba se ujistit, že text je na dané barvě pozadí čitelný [38].

### ■ 3.11.2 Popis některých komponent

#### ■ Tlačítka

Tlačítka v aplikaci mohou být tři typů - flat button (ploché tlačítko), raised button (vyvýšené tlačítko) a floating action button (plovoucí akční tlačítko). Všechna tlačítka se po stisknutí vyplní barvou.

Ploché tlačítko nemá žádný zdvih, zobrazuje se tedy přímo na materiálu jako jeho součást. Používají se často v dialogích či app barech.

Vyvýšená tlačítka jsou standardně obdélníková a mají definovaný zdvih 2dp [39].

Plovoucí akční tlačítka jsou kulatá a mají definovaný zdvih větší, než tlačítka vyvýšená. Plovoucí tlačítka jsou určena pro nejdůležitější akce na dané obrazovce a nedoporučuje se používat více těchto tlačítek na jedné obrazovce zároveň. Po stisknutí může obsahovat další tlačítka s akcemi [40].

#### ■ Navigation drawer

Navigation drawer obsahuje hlavní navigaci aplikace, která se dá otevřít přetažením prstu po obrazovce zleva doprava. Má definovaný zdvih 16dp a otevřený částečně překrývá obrazovku, přičemž viditelná část obrazovky je ztmavena.

Navigation drawer může být i permanentní, tedy stále viditelný a zakotvený na levé straně, a tento přístup se doporučuje pro desktopové aplikace. Dočasný navigation drawer, který je možné otevírat a zavírat, je naopak doporučovaný pro tablety a vyžadovaný pro mobilní telefony [41].



## ■ Dialogová okna

Dialogová okna mohou být různých typů:

- *Upozornění* - "naléhavá vyrušení, která potřebují potvrzení a informují uživatele o situaci" [42]
- *Jednoduché dialogy* - mohou poskytovat doplňující detaily či akce a neobsahují tlačítka pro potvrzení či zrušení
- *Potvrzující dialogy* - vyžadují, aby uživatel svou volbu potvrdil pomocí tlačítka, které dialog obsahuje společně s tlačítkem pro zrušení dialogu; může se jednat o výběr ze seznamu, ale také výběr data či času
- *Dialogy přes celou obrazovku* - používají se jen na mobilních telefonech, kde je limitovaný prostor, a mohou mít složitější layout, ve kterém je možné otevřít jednoduchý dialog jako součást složitější operace [42]





## Část II

### Praktická část

## Kapitola 4

### Vlastní aplikace

Aplikaci, pojmenovanou Power Supply Remote Manson, jsem vytvořila ve vývojovém prostředí Android Studio v programovacím jazyce Java. Minimální podporovanou verzí je 4.1 Jelly Bean (úroveň API 16). Vzhledem k tomu, že starší verze se již téměř nepoužívají, je možné aplikaci spustit na téměř všech zařízeních s OS Android.

Grafický návrh aplikace využívá principů Material designu, přičemž primární barva je modrozelená a byla zvolena tak, aby ladila s barvou používanou firmou Manson. Jako doplňkovou barvu jsem pak volila broskvově oranžovou, jelikož společně s modrozelenou působí vizuálně příjemně. Layout je přizpůsoben různým rozměrům i orientacím obrazovky a po spuštění si aplikace ponechává výchozí orientaci, což znamená, že s otočením zařízení o 90° nedojde k automatickému otočení obrazu.

Aplikace se po připojení ke zdroji spouští automaticky. Toho je docíleno pomocí filtru záměru (intent filter) pro připojení USB zařízení a XML souboru, kde je specifikováno Vendor ID.

#### 4.1 Ikona

Ikonu aplikace, která je na obrázku 4.1, jsem vytvořila pomocí programu Adobe Photoshop Elements 7.0 a je laděna do modrozelené barvy využívané aplikací. Dolní polovina obrázku graficky znázorňuje zdroj při pohledu zepředu, obdélník s písmenem *M* v pravém horním rohu představuje mobilní telefon s aplikací pro ovládání zdroje a šipky pak znázorňují komunikaci mezi mobilním zařízením a zdrojem.



**Obrázek 4.1:** Ikona vlastní aplikace

## 4.2 Vymezení pojmů program a profil v kontextu vytvořené aplikace

Programem se v kontextu aplikace myslí uživatelem vytvořená sekvence hodnot napětí a proudu společně s časem, po jaký mají být tyto hodnoty na zdroji nastaveny. Uživatelem zadaná hodnota napětí, proudu a času představuje jednu položku programu, přičemž počet těchto položek, neboli změn hodnot na zdroji, není omezen.

Profilem je myšlena jediná hodnota napětí a proudu, která se na zdroji nastaví po nedefinovanou dobu - nastavení se tedy nemění do další manuální změny hodnot napětí a proudu.

## 4.3 Struktura

Aplikace má jednu aktivitu (MainActivity), která se spouští při startu a obsahuje všechny fragmenty aplikace, které se přidávají a odebírají na základě interakce s uživatelem.

Seznam fragmentů:

- *MainFragment* - základní ovládání zdroje
- *ProfilesFragment* - seznam vytvořených profilů
- *AddFragment* - přidání nového profilu
- *UpperLimitFragment* - nastavení horního limitu napětí a proudu
- *ProgramListFragment* - seznam vytvořených programů
- *AddProgramFragment* - vytvoření nového programu či úprava starého a jeho uložení
- *ProgramDetailFragment* - detail programu včetně jeho spuštění
- *GraphFragment* - zobrazení grafu
- *StatusListFragment* - zobrazení dat v čase v textové podobě
- *SettingsFragment* - nastavení aplikace

### 4.3.1 MainFragment

Jedná se o hlavní fragment, který se zobrazí při spuštění aplikace. Název v app baru je totožný s názvem aplikace, tedy "*Power Supply Remote Manson*". S celým UI fragmentu je možné interagovat jen za předpokladu, že je aplikace spojena se zdrojem.

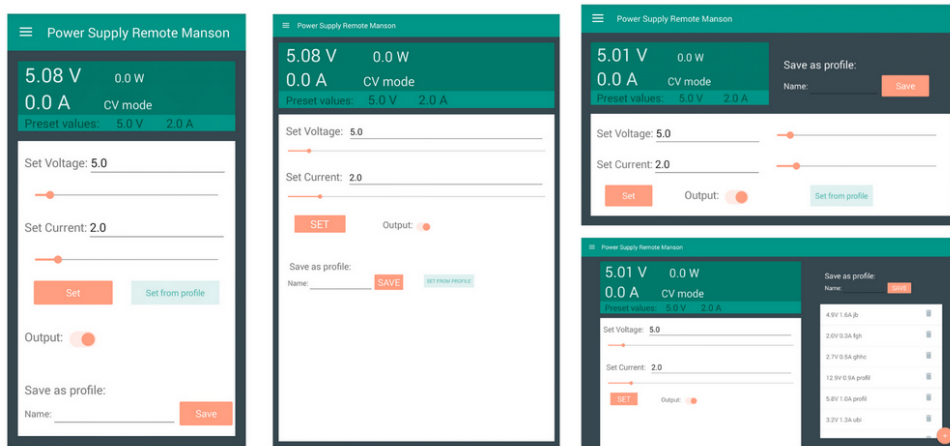
V horní části obrazovky je zobrazen virtuální displej, kde, pokud je aplikace spojena se zdrojem, jsou každou sekundu aktualizovány hodnoty napětí, proudu, výkonu, režimu konstantního proudu či konstantního napětí, nastaveného napětí a nastaveného proudu na zdroji. V případě, že aplikace se zdrojem spojena není, místo módu se na displeji zobrazí hláška

"No connection!", případně "Connecting...", pokud právě dochází k připojení.

Z tohoto fragmentu může uživatel nastavit na zdroji hodnoty napětí a proudu a také vypnout či zapnout výstup zdroje (zda je na výstupních svorkách napětí). Hodnoty napětí a proudu, které chce na zdroji nastavit, může buď zadat do editačního okna, nebo vybrat pomocí posuvníku. V obou případech ovšem nemůže zadat hodnotu vyšší, než je maximální výstupní hodnota zdroje. Editační okno je s příslušným posuvníkem spárováno tak, že při změně hodnoty v editačním okně se změní poloha posuvníku a pokud uživatel změní polohu posuvníku, naopak se změní hodnota zobrazená v editačním okně. Zapnutí a vypnutí výstupu je možné pomocí přepínače, který využívá knihovnu SwitchButton[43] pro modernější vzhled přepínače i na starších verzích OS.

Dále je možné v tomto fragmentu uložit aktuálně nastavené hodnoty na zdroj jako profil. Tak je možné učinit pomocí editačního okna, do kterého se píše název profilu, který je nepovinný, a tlačítka, kterým se nový profil uloží do paměti telefonu. Všechny profily je pak možné zobrazit v ProfilesFragmentu, který se zobrazí po stisknutí tlačítka "Set from profile", případně na stejné obrazovce po pravé straně, pokud uživatel používá zařízení s větší velikostí obrazovky v orientaci na šířku (viz dále).

Odlišný layout je poskytnut pro 5 různých konfigurací obrazovky. Tyto různé layouts jsou vidět na obrázku 4.2. Virtuální displej se vždy nachází v horní části displeje a zabírá buď celou jeho šířku, nebo se nachází na jeho levé polovině. Pokud aplikace nemá dostupný prostor na výšku alespoň 540dp, roluje se ve vertikálním směru celé UI, pokud je dostupná výška minimálně 540dp, pak je rolována jen část pod virtuálním displejem, který zůstává ukotven v horní části obrazovky.



**Obrázek 4.2:** MainFragment na různých konfiguracích obrazovky

Drobné změny v UI mají zařízení s minimální dostupnou šířkou obrazovky 560dp při orientaci na výšku. Pověšmout si lze jen nevýrazných úprav v pozicích jednotlivých elementů.

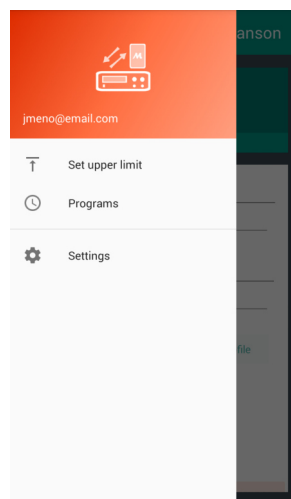
Zajímavější změny jsou patrné při orientaci zařízení na šířku. Zde je možné

si povšimnout snahy využít šířku displeje a naopak omezit výšku UI. Na zařízeních s dostupnou šířkou obrazovky mezi 540dp a 840dp je napravo od displeje umístěno ukládání nastavených hodnot jako profil. Jelikož tato část fragmentu patří mezi ty méně důležité, zobrazuje se namísto bílého pozadí na pozadí tmavším, čímž na sebe upoutává méně pozornosti. Posuvník pro nastavení hodnoty napětí a proudu se zobrazuje vedle editačního okna a v jedné řadě se též zobrazují zbylá tlačítka včetně přepínače.

Ze všech layoutů je pravděpodobně nejzajímavějším layout určený pro zařízení s dostupnou šířkou minimálně 840dp při orientaci na šířku. Po spuštění aplikace se zde na obrazovce totiž zobrazí 2 fragmenty zároveň. Levá polovina obrazovky je určena MainFragmentu, ve kterém chybí tlačítko pro zobrazení fragmentu se seznamem profilů, jelikož tento fragment se zobrazuje na pravé polovině obrazovky a žádné tlačítko tedy není potřeba. Ukládání nastavených hodnot do profilu je pak zobrazeno nad ProfilesFragmentem na pravé straně od virtuálního displeje.

Z fragmentu je též dostupný Navigation drawer, prostřednictvím kterého se dají zobrazit další fragmenty aplikace. Ve vrchní části, s pozadím v odstínech doplňkové barvy, se zobrazuje obrázek bílé ikony aplikace a pod ní je napsána e-mailová adresa uživatele, pokud ji zadal ve správném formátu v nastavení aplikace.

Pod ní se nachází třípoložkové menu s položkami "*Set upper limit*" pro spuštění UpperLimitFragmentu, "*Programs*" pro zobrazení fragmentu ProgramListFragment a "*Settings*" pro spuštění SettingsFragmentu. Vedle každé položky se zobrazuje i ikona, která znázorňuje příslušnou funkci.



Obrázek 4.3: Navigation drawer

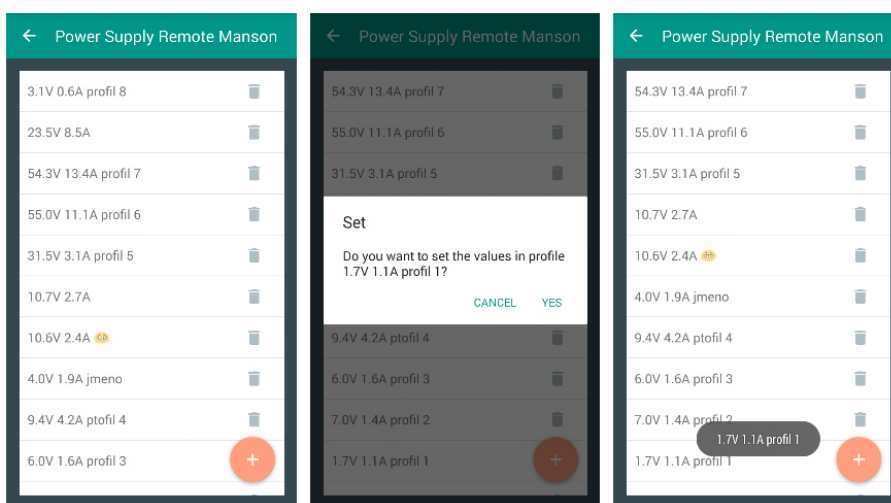
### 4.3.2 ProfilesFragment

ProfilesFragment zobrazuje ListView se seznamem vytvořených profilů. V pravém dolním rohu se nachází floating button, uvnitř kterého je znak plus a po jehož stisknutí se zobrazí AddFragment. Fragment má pro všechny konfigurace obrazovky stejný layout.

Seznam fragmentů je nejprve načten ze souboru a následně zobrazen v seznamu pomocí adaptéru, který mimo jiné definuje, jak budou jednotlivé položky seznamu vypadat. Každá položka obsahuje text, ve kterém jsou napsány hodnoty napětí a proudu uložené v daném profilu a také název daného profilu, pokud nějaký má. V pravém rohu každé položky je pak ikona koše, po jejímž stisknutí se na obrazovce zobrazí potvrzovací dialogové okno s otázkou, zda chce uživatel příslušný profil opravdu smazat. Pokud svou

volbu potvrdí, daný profil zmizí ze seznamu a soubor s uloženými profily je přepsán.

Hodnoty z konkrétního profilu je na zdroj možné nastavit stisknutím příslušné položky v seznamu. I při nastavení hodnot na zdroj se zobrazí dialogové okno, kterým uživatel svou volbu buď potvrdí, nebo zruší. Pokud je aplikace spojena se zdrojem a nastavované hodnoty nejsou větší, než maximální hodnoty zdroje, jsou hodnoty na zdroji nastaveny a na displeji se zobrazí krátká zpráva o jejich nastavení. V opačném případě se na obrazovce buď zobrazí zpráva, že zdroj není připojen, nebo zpráva, že hodnoty přesahují maximální limit.

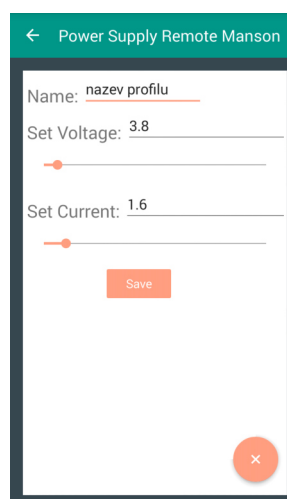


Obrázek 4.4: ProfilesFragment s příkladem nastavení profilu na zdroj

### 4.3.3 AddFragment

Účelem tohoto fragmentu je přidání nového profilu do seznamu uložených profilů. UI je pro všechny konfigurace obrazovky totožné a sestává z editačních oken pro vepsání hodnot napětí a proudu, které chce uživatel v profilu nastavit, a nepovinného názvu profilu. Hodnoty je možné nastavovat i pomocí posuvníku, který se mění v závislosti na hodnotě editačního okna a naopak. Profil je pak možné uložit pomocí tlačítka "Save".

V pravém dolním rohu se nachází plovoucí tlačítko se znakem křížku uvnitř. To slouží k návratu k předchozímu fragmentu se seznamem profilů. Navíc obsahuje i animaci, kdy po jeho stisknutí se znak uvnitř otočí o 45°, čímž se mění na plus či křížek v závislosti na tom, jaký fragment je v danou chvíli zobrazen.



Obrázek 4.5: AddFragment

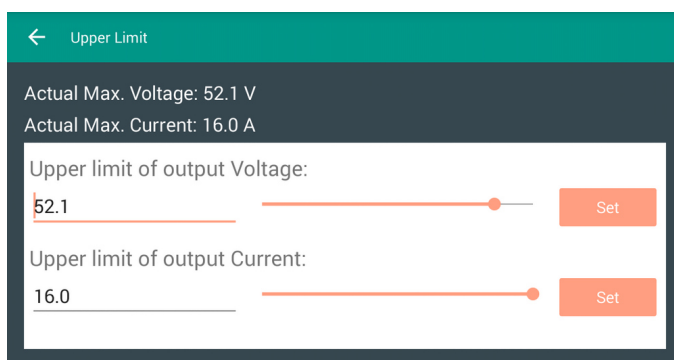


### 4.3.4 UpperLimitFragment

Tento fragment slouží k nastavení maximálního výstupního napětí a proudu. V horní část se v textovém poli zobrazují aktuálně nastavené maximální hodnoty. Uživatel má možnost tyto hodnoty změnit prostřednictvím příslušného posuvníku či editačního okna, přičemž minimální možnou hodnotou může být 1, jelikož se jedná o minimální hodnotu, kterou je na zdroji možné nastavit pro maximální výstupní proud či napětí, a maximální možnou hodnotou je maximální hodnota napětí či proudu, které je zdroj schopen. Tato hodnota se dá zjistit pomocí speciálního příkazu poslaného na zdroj.

Příkaz na změnu maximálního výstupního napětí či proudu je aplikací poslán po stisknutí tlačítka "Set" u příslušné veličiny.

Layout se na různých konfiguracích obrazovky příliš nemění. Při orientaci na výšku se posuvník zobrazuje pod EditTextem, kdežto při orientaci na šířku se zobrazuje vedle něj.



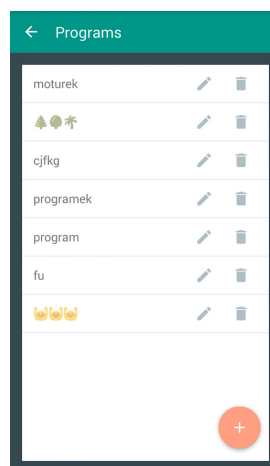
Obrázek 4.6: UpperLimitFragment při orientaci zařízení na šířku

### 4.3.5 ProgramListFragment

V tomto fragmentu se zobrazuje seznam uložených programů, které jsou načteny ze souboru, a jeho layout se zobrazuje na všech zařízeních stejně. V app baru se zobrazuje název "Programs".

Jednotlivé programy se pomocí adaptéru zobrazí v ListViewu, ve kterém je možné vertikálně rolovat. Každá položka obsahuje název programu a v pravé části ikonu tužky, která spustí fragment AddProgramFragment pro editaci programu, a koše pro smazání příslušného programu, které musí uživatel potvrdit v zobrazeném dialogovém okně.

V pravém dolním rohu se zobrazuje floating button, uvnitř nějž je znak plus. Stiskem tohoto tlačítka se vytvoří fragment AddProgramFragment pro přidání nového programu.



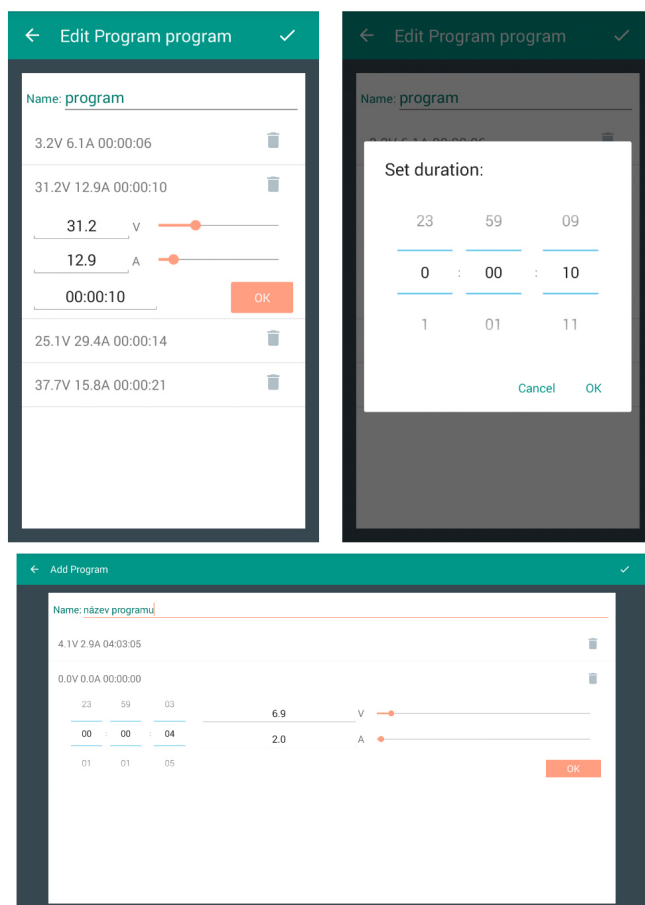
Obrázek 4.7: Layout ProgramListFragmentu

### 4.3.6 AddProgramFragment

AddProgramFragment, jehož layout je vyborazen na obrázku 4.8, slouží buď k vytvoření nového programu, nebo pro úpravu již existujícího programu. Podle toho se mění i název zobrazení v app baru, který je buď "Add Program", nebo "Edit Program" s názvem upravovaného programu. V pravé části app baru je pak ikona fajfky, kterou je program uložen.

Pod app barem se nachází editační okno, do kterého se zadává název programu, který je povinný a v případě jeho nevyplnění se na obrazovce zobrazí zpráva, že je jméno prázdné.

Zbytek obrazovky zabírá ListView, který obsahuje jednotlivé položky programu. V každé položce se zobrazují v ní nastavené hodnoty napětí, proudu a času, který je ve formátu hh:mm:ss. V pravé části se nachází ikona koše, která umožňuje vybranou položku smazat, přičemž její smazání je nutné potvrdit v dialogovém okně. Při vytvoření nového programu se v seznamu na počátku zobrazí pouze jedna položka se všemi hodnotami rovnými 0.



**Obrázek 4.8:** AddProgramFragment spolu se zadáváním doby trvání na různých obrazovkách

Po stisknutí položky se pod ní zobrazí editační okna s posuvníky pro nastavení napětí a proudu a okno, po jehož stisknutí se zobrazí dialog s možností nastavit dobu trvání. Výběr času je realizován pomocí `TimePickerů` a umožňuje nastavit sekundy a minuty v intervalu 0 až 59 a hodiny v rozmezí 0 až 23. Pokud uživatel již dříve dobu trvání nastavoval, nastaví se v dialogu tyto hodnoty jako výchozí, což usnadňuje změnu. Díky většímu místu v horizontálním směru se na větších obrazovkách výběr času nezobrazuje v dialogovém okně, ale `TimePickery` jsou umístěny nalevo od editačních oken pro změnu napětí a proudu. Do položky se uživatelem nastavené hodnoty uloží stisknutím tlačítka "OK". Pokud je editovaná položka v seznamu poslední, po stisknutí tlačítka "OK" se vytvoří nová položka se všemi hodnotami rovnými 0. Editovat je možné vždy pouze jednu položku.

Při ukládání je z programu smazána poslední položka seznamu, pokud obsahuje pouze nulové hodnoty. Pokud je tato položka zároveň jedinou v seznamu, zobrazí se na obrazovce zpráva, že program je prázdný. Při uložení položek programu do textového souboru je také aktualizován soubor se seznamem všech programů, kam je přidán název programu spolu s unikátním číslem, které je obsaženo i v názvu souboru uloženého programu. Při výběru programu v části `ProgramListFragment` je pak zobrazen detail vybraného programu načtením dat ze souboru uloženého s tímto unikátním číslem.

### 4.3.7 ProgramDetailFragment

Tento fragment slouží pro spuštění vytvořeného programu. `App bar` nese název "Program" spolu s názvem vybraného programu a ikonu tužky pro spuštění `AddProgramFragmentu` pro editaci programu.

Stejně jako u `MainFragmentu`, i zde je zobrazen virtuální displej, kde se každou sekundu aktualizují hodnoty a kde je navíc zobrazena hodnota elektrického náboje v ampérhodinách a naopak v něm chybí hodnoty nastavené na zdroji.

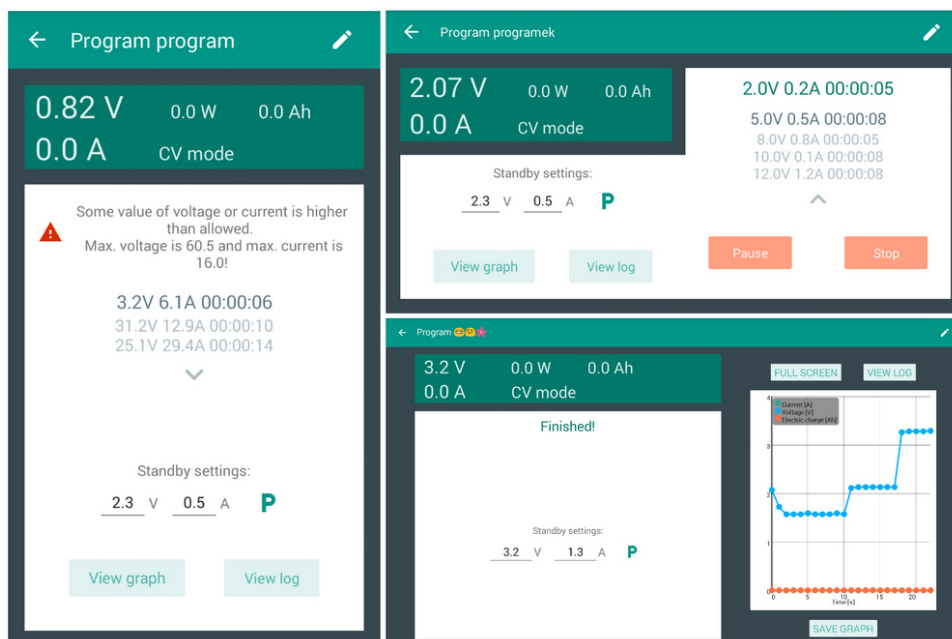
Hodnoty napětí a proudu, které jsou právě nastaveny na zdroji, se spolu se zbývajícím časem zobrazují nad seznamem zbývajících položek programu. Aktuálně nastavená hodnota je zobrazena nejvýše a má největší velikost písma a poutavou modrozelenou barvu, následující položka má menší, tmavě šedé písmo a zobrazuje se pod ní. Zbýlé položky pod nimi mají světle šedou barvu a nejmenší písmo. Po skončení jedné položky se na zdroj pošlou hodnoty napětí a proudu nadcházejícího programu a předchozí položka zmizí. Zobrazený seznam se tedy celý posune o jedno místo nahoru a o jednu položku se zkrátí. Pokud zobrazený seznam obsahuje více než 3 položky, je pod ním zobrazena šipka, která umožňuje seznam rozbalit či zabalit tak, že se buď zobrazuje celý, nebo se zobrazují pouze jeho první 3 položky. Po skončení programu se místo první položky vypíše text "Finished!".

Pokud program není spuštěn, zobrazuje se nad seznamem tlačítko "Start" pro jeho spuštění, případně upozornění, že aplikace není připojena ke zdroji,

nebo že některá položka ze seznamu obsahuje hodnotu napětí či proudu větší, než je maximální limit příslušné veličiny. Pokud program naopak spuštěn je, zobrazí se pod seznamem položek tlačítka pro jeho pozastavení či zastavení. Při pozastavení stisknutím tlačítka "Pause" se zastaví odpočítávání času pro aktuálně nastavené hodnoty. Tlačítko změní svou barvu z oranžové na zelenou a umožní pak program opět spustit tam, kde byl pozastaven. Zastavením programu je celý běh programu ukončen, tlačítka "Pause" a "Stop" zmizí a zobrazí se tlačítko "Start" nad seznamem, ve kterém jsou vypsané všechny položky daného programu.

Fragment dále obsahuje část nazvanou "Standby settings", ve které jsou dvě editační okna pro nastavení hodnoty napětí a proudu, které se na zdroj pošlou v případě, že je program pozastaven, zastaven či ukončen. Napravo od editačních oken je ikona s písmenem "P", která umožňuje do editačních oken dosadit hodnoty z profilu, který uživatel vybere prostřednictvím zobrazeného dialogového okna.

Průběh napětí, proudu a elektrického náboje v závislosti na čase je možné zobrazit uvnitř fragmentů GraphFragment a StatusListFragment, které jsou dostupné prostřednictvím příslušného tlačítka. Tyto hodnoty se zároveň s každou aktualizací virtuálního displeje ukládají do souboru ve vnitřní paměti zařízení, odkud jsou dočasně přístupné aplikaci, která je dále zobrazuje v grafu či v seznamu.



**Obrázek 4.9:** ProgramDetailFragment na různých konfiguracích obrazovky

ProgramDetailFragment, jehož tři různé layouty lze vidět na obrázku 4.9, má pravděpodobně nejsložitější UI v celé aplikaci. Ve verzi pro nejmenší zařízení jsou všechny části UI fragmentu zobrazeny pod sebou a celé UI, včetně virtuálního displeje, je možné vertikálně rolovat. Totožný layout se zobrazí i na zařízeních s větší velikostí obrazovky v orientaci na výšku, jen

virtuální displej zůstává ukotven v horní části obrazovky a vertikálně rolována je pouze zbylá část UI pod ním.

Větší změny v layoutu jsou patrné při orientaci zařízení na šířku. Pokud je dostupná šířka zařízení mezi 540dp a 840dp, část UI se seznamem položek programu a tlačítka k jeho ovládní se zobrazuje na pravé polovině obrazovky. Virtuální displej je pak umístěn v levém horním rohu a zbytek UI se nachází pod ním.

Pokud je dostupná šířka obrazovky při orientaci na šířku alespoň 840dp, zobrazí se na levé straně obrazovky pod sebou virtuální displej, část se seznamem programů a část "*Standby settings*". Na pravé straně obrazovky je pak `FrameLayout`, uvnitř kterého je zobrazen buď `GraphFragment`, nebo `StatusListFragment`. Mezi těmito dvěma fragmenty může uživatel přepínat pomocí tlačítek nad nimi. Fragment `GraphFragment` lze také pomocí tlačítka zobrazit přes celou obrazovku. Graf či data v souboru CSV lze uložit pomocí tlačítka, které se nachází pod fragmentem.

Program může běžet na pozadí, je tedy možné vypnout obrazovku zařízení či používat zařízení, aniž by se přerušilo odpočítávání a posílání hodnot na zdroj.

#### 4.3.8 GraphFragment

V tomto fragmentu se zobrazuje graf, který vykresluje hodnoty napětí a proudu na zdroji a vypočítaného elektrického náboje v závislosti na čase. Pro vytvoření grafu je použita open-source knihovna `GraphView`[44], která rozšiřuje třídu `View`, takže je možné `GraphView` v layoutu použít jako jakýkoliv jiný `View`.

Důležité komponenty:

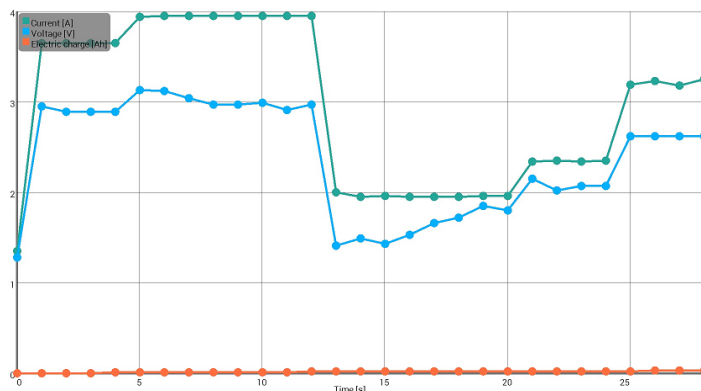
- *Series (řada)* - třída, která slouží k vytvoření dat grafu
- *Viewport (výřez)* - specifikuje část grafu, která je momentálně viditelná, přičemž je možné výřez přibližovat a posouvat, pokud jsou tyto funkce aktivovány
- *Grid Label Renderer (vykreslovač mřížky a popisků)* - vytváří mřížku a popisky horizontální a vertikální osy [45]

Vývojář si graf může přizpůsobit dle libosti. Má například na výběr z několika typů grafu (spojnicový, bodový, sloupcový), může měnit jejich barvu, tloušťku čar či velikost bodů, měnit velikost textu, přidat legendu či upravovat formát popisu os. V jednom obrázku může být vykresleno více grafů najednou [46]. Grafy v aplikaci jsou spojnicového typu a jednotlivé veličiny jsou barevně odlišeny. Napětí je zobrazeno modře, proud zeleně a elektrický náboj oranžově. Odstíny barev jsou voleny tak, aby ladily s barvami aplikace. V levém horním rohu je umístěna legenda.

Knihovna `GraphView` také umožňuje měnit a zobrazovat graf v reálném čase. Napomáhají tomu metody `resetData` (vynulování dat) a `appendData` (přidání nových dat) [47]. Díky tomu je možné sledovat průběh hodnot v čase za běhu programu. Pokud je `GraphFragment` spuštěný, dostává

aktuální data z ProgramDetailFragmentu každou vteřinu a graf je pokaždé aktualizován. Pokud GraphFragment spuštěný není, všechna data jsou do grafu doplněna při jeho spuštění.

V pravém rohu app baru se nachází ikona pro uložení grafu jako obrázku ve formátu JPEG do složky "MansonRemote" v galerii. Zároveň je obrázek poslán na e-mail, pokud jej uživatel vyplnil v nastavení.



**Obrázek 4.10:** Příklad uloženého grafu (zeleně proud [A], modře napětí [V] a oranžově elektrický náboj [Ah])

### 4.3.9 StatusListFragment

Ve StatusListFragmentu jsou hodnoty napětí, proudu a elektrického náboje pro každý bod v čase zobrazeny v textové podobě uvnitř seznamu, který vizuálně tvoří tabulku, v jejímž prvním sloupci je napsán čas ve formátu hh:mm:ss, následuje hodnota napětí, dále hodnota proudu a v posledním sloupci hodnota elektrického náboje. Text každé veličiny má totožnou barvu jako křivka příslušné veličiny v GraphFragmentu.

Hodnoty se vypisují do seznamu postupně, první položka tedy obsahuje hodnoty v čase 0 a poslední položka naposledy získané hodnoty. Pokud je fragment zobrazen za běhu programu, je seznam aktualizován každou sekundu při získání aktuálních hodnot ze zdroje.

V pravém rohu app baru se nachází ikona pro uložení seznamu, s jednotlivými hodnotami oddělenými středníky, do souboru ve formátu CSV, jehož názvem je jméno programu společně s aktuálním datem a časem při uložení. Vytvořený soubor je následně dostupný ve složce "MansonRemote" mezi dokumenty. Zároveň je i odeslán e-mail, pokud jej uživatel vyplnil v nastavení. Podle preferencí uživatele je e-mailem buď odeslán CSV soubor, nebo tabulka vytvořená v HTML uvnitř těla e-mailu.

Time [hh:mm:ss]	Voltage [V]	Current [A]	Electric Charge [Ah]
00:00:07	2.07V	0.0A	0.0Ah
00:00:08	2.07V	0.0A	0.0Ah
00:00:09	2.07V	0.0A	0.0Ah
00:00:10	2.07V	0.0A	0.0Ah
00:00:11	4.57V	0.0A	0.0Ah
00:00:12	5.07V	0.0A	0.0Ah
00:00:13	5.08V	0.0A	0.0Ah
00:00:14	5.01V	0.0A	0.0Ah
00:00:15	5.08V	0.0A	0.0Ah
00:00:16	5.08V	0.0A	0.0Ah

**Obrázek 4.11:** Příklad StatusListFragmentu

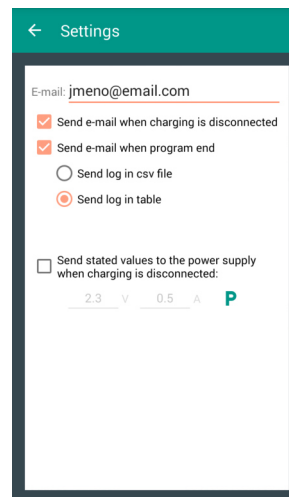
### 4.3.10 SettingsFragment

V tomto fragmentu si uživatel může nastavit své preference na základě toho, jak chce aplikaci využívat. Jeho UI je možné vidět na obrázku 4.12. Nachází se zde editační okno pro vyplnění e-mailové adresy uživatele, které automaticky kontroluje formát vyplněného e-mailu a hlásí chybu, pokud je jeho formát neplatný. Vyplněná e-mailová adresa udává, kam se mají posílat zprávy související s aplikací.

Uživatel si sám vybere, v jakých situacích chce dostat zprávu, pomocí zaškrťvacích polí, ve kterých si může vybrat, zda chce obdržet zprávu při výpadku napájení zařízení a při skončení programu. Pokud si přeje získat zprávu při skončení programu, má možnost si zvolit, v jakém formátu mu budou odeslána data o jeho průběhu v čase - buď jsou poslána ve formátu CSV, nebo v těle zprávy uvnitř tabulky vytvořené v HTML. Ve vybraném formátu jsou posílána i data ze StatusListFragmentu při ukládání, kdy se data na e-mail posílají vždy, když je e-mail v nastavení vyplněn.

Poslední funkcí, kterou uživatel může nastavit, je možnost posílání definovaných hodnot napětí a proudu na zdroj v případě, že je odpojeno napájení zařízení. Pokud tuto možnost uživatel zaškrtně, pak v editačních polích může specifikovat hodnoty, které lze vybrat i ze seznamu profilů zobrazených v dialogovém okně, prostřednictvím ikony s písmenem "P". Pokud jsou hodnoty vyplněny, nastaví se automaticky i do editačních polí v části "Standby settings" ve ProgramDetailFragmentu.

Celé nastavení je po volání metody `onDestroy` fragmentu uloženo do souboru, odkud je dostupné. Při každém spuštění fragmentu je uložené nastavení z tohoto souboru načteno a zobrazeno.



Obrázek 4.12: Layout SettingsFragmentu

## 4.4 Funkce

Aplikace má následující funkce:

- nastavení hodnoty napětí a proudu na zdroji
- zapnutí/vypnutí výstupu zdroje
- virtuální displej zdroje (zobrazí aktuálních hodnot napětí, proudu a výkonu, režimu konstantního napětí či konstantního proudu a aktuálně nastavených hodnot napětí a proudu)
- vytvoření profilu a nastavení jeho hodnot na zdroji
- nastavení horního limitu napětí a proudu
- vytvoření programu a jeho spuštění

- nastavení hodnot napětí a proudu poslaných na zdroj v případě přerušení programu či výpadku síťového napájení
- zobrazení průběhu programu v grafické či textové podobě (i v reálném čase)
- uložení průběhu programu do souboru na externí úložiště jako obrázek ve formátu JPEG nebo v do souboru CSV pro zobrazení v tabulce
- posílání zpráv na e-mail
- hlídání výpadku síťového napájení

#### 4.4.1 Komunikace se zdrojem

Pro komunikaci se zdrojem aplikace vyžaduje funkci USB host, která není podporována všemi zařízeními s OS Android. Z tohoto důvodu je v manifestu deklarováno, že aplikace tuto funkci využívá.

Komunikace je realizována s využitím knihovny UsbSerial[48] pro jednodušší správu komunikace. Knihovna podporuje zařízení s čipem CP210x firmy Silicon Labs, který používá zdroj HCS-3604-USB pro převod z USB na sériový port. Nejstarší knihovnou podporovaná verze Androidu je 3.1 (Honeycomb API 12), což pokrývá všechny verze podporované aplikací.

Pro komunikaci se zdrojem (přijímání a posílání dat) knihovna používá 2 buffery o velikosti 16kb a pro přijímání dat je definován callback, přičemž na zařízeních s úrovní API 17 a vyšší je tato operace realizována asynchronně. Zapisování je vždy realizováno jako synchronní operace [49].

Komunikace se zdrojem je spravována v aktivitě MainActivity, v níž se spouští všechny fragmenty. Při připojení USB zařízení aplikace nejprve zjistí podle jeho Vendor ID, zda se jedná o hledané zařízení. Následně aplikace požádá uživatele o povolení k připojení k danému zařízení. Uživatel toto nastavení může uložit jako výchozí, aby se toto okno již nezobrazovalo a zařízení dostalo povolení k připojení automaticky. Vzhledem k tomu, že aplikace využívá filtry záměrů pro automatické spuštění aplikace, zařízení též získá povolení automaticky v případě, kdy uživatel vybere tuto aplikaci pro zpracování tohoto záměru [50].

Pokud je komunikace se zařízením povolena, otevře se sériový port, nastaví se parametry této komunikace (dle nastavení v kapitole 2.5), je definován callback pro přijímání dat ze zdroje a aktuálně zobrazený fragment je informován o připojení zařízení, pokud tuto informaci pro své fungování potřebuje. Při připojení zdroje v MainFragmentu je například na zdroj poslána série příkazů pro zjištění horního limitu, maximálních hodnot a nastavených hodnot napětí a proudu a hodnot napětí, proudu a režimu na displeji.

Posílání dat na zdroj je uskutečněno uvnitř synchronizované metody, která zajistí, že je na zdroj v jednu chvíli poslán jediný příkaz, a počká na jeho vyřízení. Příkazy, které je možné na zdroj poslat jsou specifikovány v tabulce A.1 v příloze. Na přečtené bajty poslané ze zdroje je aplikováno kódování UTF-8 a zpráva je vepsána do neviditelného textového pole



v aplikaci, odkud ji lze dále číst a zpracovat.

Při odpojení zdroje od zařízení je sériový port uzavřen a na tuto skutečnost je upozorněn i spuštěný fragment.

#### 4.4.2 Ukládání do souboru

Pro ukládání do souboru je nutné deklarovat v manifestu oprávnění pro zápis do externího úložiště, čímž aplikace zároveň získá i oprávnění pro jeho čtení [51].

Textové soubory obsahující uživatelské nastavení aplikace nebo seznam profilů či programů jsou aplikací ukládány ve vnitřním úložišti do složky aplikace a nejsou pro uživatele viditelné. Tato složka je určena názvem balíčku aplikace [51].

Soubory průběhu programů, které uživatel ukládá manuálně, jsou ukládány do externího úložiště jako veřejně přístupné, jsou tedy viditelné pro uživatele a dostupné i po odinstalování aplikace [51], přičemž CSV soubory jsou ukládány do adresáře s dokumenty a obrázky grafů, komprimované do formátu JPEG, do adresáře s obrázky. V obou případech se soubory pro větší přehlednost ukládají do podsložky nazvané "*MansonRemote*".

#### 4.4.3 Posílání zpráv na e-mail

Pro posílání zpráv na e-mail, pokud je zadaný uživatelem v nastavení, byla vytvořena e-mailová adresa "*mansonremoteapp@gmail.com*". Pro posílání zpráv je využita portovaná verze JavaMail API pro Android[52], přičemž pro přenos je použit protokol SMTP (Simple Mail Transfer Protocol).

Hlavní třídou JavaMail API je *Session*, která zpracovává nastavení konfigurace a autentizaci. V ní jsou nastaveny základní informace (název poštovního serveru, číslo portu, uživatelské jméno, heslo a další). Pro vytvoření zprávy slouží třída *MimeMessage*, která je podtřídou třídy *Message* a v jejímž konstruktoru musí být předán vytvořený objekt *Session*. Pomocí metod této třídy je pak nastaven odesílatel, příjemce, předmět a tělo či příloha zprávy. Pro samotné poslání zprávy pak slouží třída *Transport*, která obvykle používá SMTP protokol [53].

V aplikaci je pro odesílání e-mailů vytvořena metoda uvnitř *MainActivity*. *Fragment*, ze kterého se zpráva posílá, volá metodu v této aktivitě. Pokud je v nastavení uvedena e-mailová adresa ve správném formátu, je vytvořeno nové vlákno, ve kterém je zpráva poslána, přičemž tento proces běží na pozadí a uživatel tedy při své interakci s aplikací nemusí čekat na to, až se zpráva pošle.

V manifestu musí být uvedeno oprávnění k použití internetu a oprávnění pro zjištění stavu sítě. Díky tomu může aplikace získat informace o tom, zda je uživatel připojen k internetu, a následně může e-mail odeslat. Pokud uživatel není připojen k internetu, zobrazí se na obrazovce hláška, že e-mail

nebyl odeslán společně s výzvou pro zkontrolování připojení k internetu. Pokud je e-mail úspěšně poslán, zobrazí se hláška o jeho odeslání.

#### ■ 4.4.4 Detekce výpadku síťového napájení

Pro detekci napájení v aplikaci slouží broadcast receiver, který získává informace o stavu baterie. Aplikace uvnitř metody onReceive, která je volána při každé změně stavu baterie, zjišťuje, zda se baterie nabíjí, její teplotu a úroveň jejího nabití.

Za předpokladu, že je aplikace připojena ke zdroji a uživatel v nastavení zatrhl posílání e-mailu či odeslání definovaných hodnot na zdroj při výpadku napájení, jsou implementovány 3 případy, které mohou nastat:

- baterie se nabíjí - na displeji se zobrazí hláška *"Battery is charging"*
- baterie se nenabíjí, její teplota je vyšší než 42°C a úroveň nabití je vyšší než 50% - na displeji se zobrazí krátká zpráva s informací o aktuální teplotě baterie
- baterie se nenabíjí, ale má nižší teplotu či úroveň nabití než u předchozího případu - na displeji se zobrazí hláška *"Battery is not charging"* a pokud tak uživatel v aplikaci nastavil, je na zadaný e-mail poslána informace o výpadku napájení a na zdroj jsou nastaveny hodnoty napětí a proudu definované v nastavení

Teplota baterie je hlídána pouze v případě nabíjení zařízení pomocí bezdrátové nabíječky. Při napájení zařízení bezdrátovou nabíječkou totiž může dojít k přehřátí baterie, kdy se baterie přestane nabíjet, dokud její teplota opět neklesne. Navíc, pokud je úroveň baterie alespoň 90%, není při bezdrátovém nabíjení detekován výpadek napájení, jelikož výrobce tvrdí, že při bezdrátovém nabíjení nemusí být baterie dobita na 100%.

O možnosti nabíjení zařízení při komunikaci se zdrojem dále pojednává kapitola 5.

## ■ 4.5 Porovnání s oficiální aplikací

### ■ 4.5.1 UI a podpora zařízení

Oficiální aplikace má příjemné a intuitivní GUI (grafické uživatelské rozhraní). Omezením je, že aplikace může být nainstalována pouze na zařízení s větší velikostí obrazovky, nefunguje tedy na mobilních telefonech, ale pouze na tabletech, příp. phabletech. Aplikaci je možné využívat jen při orientaci zařízení na šířku, což je standardní orientace používaná na tabletech. Tato orientace je tedy ve většině případů dostačující. Minimální podporovanou verzí je Android 4.0 (Ice Cream Sandwich) [3].

Vlastní aplikace využívá principů Material designu. Na rozdíl od oficiální aplikace podporuje zařízení s různou velikostí obrazovky, je ji tedy možné používat i na mobilních telefonech. Navíc lze zařízení použít v libovolné

orientaci, ačkoliv ji není možné v průběhu používání aplikace měnit. Minimální podporovanou verzí je Android 4.1 (Jelly Bean), oficiální aplikace tedy podporuje o necelé 1% více zařízení [15].

### ■ 4.5.2 Komunikace a napájení

K oficiální aplikaci je pro komunikaci se zdrojem nutné vlastnit Raspberry Pi, na které je potřeba nainstalovat driver. Tím je samotná komunikace se zdrojem pro uživatele podstatně složitější. Výhodou tohoto přístupu je však bezdrátová komunikace, která zařízení umožňuje standardní napájení přes micro USB port. Aplikace ovšem nemá implementovanou funkci detekce výpadku síťového napájení.

Vlastní aplikace komunikuje přes USB a uživatel si tak pro komunikaci se zdrojem vystačí pouze s USB OTG (On-The-Go) kabelem. Vzhledem k tomu, že zařízení bývají napájena přes stejný micro USB port, který je potřeba ke komunikaci, je podstatně složitější zařízení zároveň napájet, pokud to uživatel vyžaduje. Tomuto tématu se blíže věnuje kapitola 5. Aplikace dokáže detekovat, zda je baterie napájena, či nikoliv.

### ■ 4.5.3 Vytvoření programů a profilů

V oficiální aplikaci je možné vytvářet programy, samostatné vytvoření profilů zde možné není. Celý program je možné opakovat v cyklech. Hodnoty napětí a proudu jednotlivých položek jsou zobrazeny v grafu. Vzhledem k tomu, že graf není možné přiblížit, ne vždy může být patrné, jaké konkrétní hodnoty jsou na osách. Zvláště při vyšší četnosti změn v programu, kterých může být až 20, se pak graf může stát nepřehledným a uživatel tak může ztratit přehled, jaké nastavení hodnot napětí a proudu následuje.

Při mém testování programů v oficiální aplikaci nefungovalo nastavení hodnoty proudu na zdroj. V zadaném čase se měnila pouze hodnota zadaného napětí. Pro uživatele, kteří potřebují v čase měnit také hodnotu proudu nastaveného na zdroj, se tak funkce vytvoření programu stává nepoužitelnou.

Neintuitivní mi také přijde označení počtu cyklů, které začíná svůj odpočet od nuly. První cyklus je tedy označen jako nultý a k finálnímu číslu se program dopočítá až po skončení posledního cyklu. Uživatel se tak snadno může nechat zmást, kolikátým cyklem vlastně program běží.

Moje aplikace umožňuje správu programů i profilů. Není zde omezen počet položek, změn v programu tedy může být libovolně mnoho. Aplikace ovšem neumožňuje opakovat program v cyklech. Položky programu se v aplikaci zobrazují v textové podobě v seznamu pod sebou podle toho, jak jdou za sebou, přičemž se zobrazují jen položky následující a aktuálně nastavená položka na zdroji.

#### ■ 4.5.4 Grafy

Oficiální aplikace umožňuje zobrazení průběhu napětí a proudu v závislosti na čase nejen v rámci programu, ale po celou dobu, co je zdroj k zařízení připojen. Zobrazení v grafu je přehledné a graf je možné zvětšit na celou obrazovku.

Vlastní aplikace vykresluje průběh do grafu jen za běhu programu. Kromě napětí a proudu ale vykresluje též závislost elektrického náboje na čase. Graf je možné zvětšit na celou obrazovku i na tabletu a je možné graf i přiblížit.

#### ■ 4.5.5 Posílání e-mailů

Oficiální aplikace umožňuje poslat průběh napětí a proudu, který se zobrazuje v části *Monitor*, e-mailem prostřednictvím jiné aplikace na zařízení k tomu určené. Uživatel tedy musí manuálně e-mail odeslat.

Vlastní aplikace spravuje posílání e-mailů automaticky, přičemž posílá též informace o skončení programu či výpadku napájení.

#### ■ 4.5.6 Další

Při testování oficiální aplikace firmy Manson jsem zjistila, že při nastavení výchozího jazyka tabletu na češtinu aplikace nefunguje správně z důvodů nahrazení desetinné tečky desetinnou čárkou. Nelze tedy využívat tlačítka plus a mínus pro změnu hodnoty napětí či proudu v části *Device*. Tato chyba se dá eliminovat nastavením výchozího jazyka tabletu například na angličtinu.

## Kapitola 5

### Napájení a komunikace

#### 5.1 USB OTG

USB On-The-Go, neboli OTG, je součástí dodatku ke specifikaci Universal Serial Bus (USB) 2.0 představenou na konci roku 2001 [54]. Umožňuje USB zařízením, jako jsou například tablety či mobilní telefony, komunikovat s jinými USB přístroji, například klávesnicí, myší či digitálním fotoaparátem, bez nutnosti služeb osobního počítače. Zachovává standardní master-slave model USB, kde řídicí zařízení (master/hostitel) posílá příkazy dalším zařízením (slave/periférie), které jsou k nim připojeny, s tím, že OTG zařízení se dle situace může chovat jako hostitel, nebo periférie [55].

OTG zařízení mají Micro-AB port, který je kompatibilní jak s konektory micro-A, tak i s konektory micro-B [54]. OTG navíc přidává ke standardnímu USB konektoru pátý pin nazvaný ID pin, který může být buď uzemněný (A-zařízení), nebo plovoucí (B-zařízení) [55]. Pokud je do micro-AB portu zasunut micro-A konektor, má ID pin hodnotu false, jinak má hodnotu true. Díky tomu může OTG zařízení detekovat na svém micro-AB portu, zda je do něj zasunut konektor micro-A, či micro-B. Toto je schopen určit na základě hodnoty odporu vůči zemi [54]. Volba mezi rolí hostitele a periférie se odvíjí od toho, ke kterému konci je zařízení připojeno. Zařízení, které je při spuštění připojeno ke konci „A“ (A-zařízení) se chová jako výchozí hostitel. Druhé zařízení, připojené na konec „B“ (B-zařízení), se chová jako výchozí periférie.[56]

Pokud uživatel připojí zařízení ke špatnému konci pro danou operaci, tedy například připojí tiskárnu jako hostitele a tiskárna se jako hostitel nemůže chovat k připojenému digitálnímu fotoaparátu, ale fotoaparát má driver k řízení tisku, zařízení si své role vymění a tiskárna přejde do role periférie a fotky tak mohou být vytištěny bez nutnosti přepojovat kabel. Zmíněnou operaci zajišťuje protokol HNP (Host Negotiation Protocol). Předpokladem je, že obě zařízení jsou OTG zařízení [56]. Hostitelské zařízení se během aktivní relace pravidelně dotazuje periférie, zda nechce přejít do role hostitele. Protokol neprojde přes standardní USB rozbočovač, jelikož je

založen na signalizaci pomocí ID pinu, který standardní USB neobsahuje [54].

Dalším novým komunikačním protokolem, které OTG přineslo, je SRP (Session Request Protocol). A-zařízení, které po celou dobu komunikace napájí sběrnici a tedy určuje, kdy se bude komunikovat, se může přepnout do režimu spánku, pokud nedochází k přenosu dat. Protokol zajišťuje, že A-zařízení v režimu spánku může být upozorněno B-zařízením, pokud je potřeba přenášet data. [56] Tím lze kontrolovat spotřebu energie, což je pro zařízení napájená z baterie důležité. OTG může spojení nechat bez napájení, dokud jej periférie nebude vyžadovat. OTG zařízení musí umět iniciovat i reagovat na SRP. Jakékoliv A-zařízení, včetně osobního počítače, může reagovat na SRP a jakékoliv B-zařízení, včetně standardní USB periférie, může iniciovat SRP [54].

Třetím protokolem zavedeným OTG je ADP (Attach Detection Protocol), který slouží k detekci připojení či odpojení zařízení při absenci napájecího napětí na sběrnici USB. Detekce připojení či odpojení zařízení je založena na změně reaktance napájecího vodiče, který je vybit a následně je změřena doba, za kterou se opět nabije na známou hodnotu [54].

Při připojení ke standardním USB zařízením se OTG zařízení chovají jako standardní USB hostitel nebo periférie. OTG hostitelé ovšem podporují pouze určitou část periférií, která je specifikovaná v TPL (Targeted Peripheral List). Jde o seznam periférií, které určité OTG zařízení podporuje, když se chová jako hostitel a deklaruje jej výrobce daného zařízení. OTG zařízení musí poskytnout dostatek energie pro tyto produkty. To ale nemusí stačit k tomu, aby se připojilo k periférii, která na tomto seznamu uvedena není [54]. Může například čerpat dostatek energie pro používání myši či klávesnice, ale nikoliv pro energeticky náročnější periférie, například externí disk. I kdyby to bylo možné, příliš by to zatížilo baterii hostitelského zařízení. Řešením může být použití napájeného USB rozbočovače. Problémem ovšem je, že hostitelské zařízení není možné napájet, jelikož jeho micro USB port je využit pro komunikaci s periférií [57].

Specifikace USB Power Delivery 2.0 umožňuje nabíjení přes USB kabel při současném používání jiných zařízení. Pomocí vhodně navrženého napájeného USB rozbočovače, označovaného OTG Y kabel, je tedy možné zařízení napájet, jestliže zařízení tuto funkci podporuje. Bohužel, rozbočovačů, které tuto část specifikace podporují, je na trhu velmi málo. Nabízené kabely navíc nejsou všechny konfigurovány stejným způsobem a energie může být směřována jen k periférii a nikoliv k hostiteli [57]. OTG Y kabel pro napájení mobilních telefonů značky Samsung například poskytuje firma Valarm [58]. Připojení USB portu k nabíječce a jinému zařízení najednou umožňuje ACA (Accessory Charger Adapter), který obsahuje 3 porty - pro připojení OTG zařízení, pro připojení periférie a napájecí, který je určený jen pro napájení a nelze skrz něj komunikovat. ACA může být též implementován pro vytvoření dokovací stanice, kdy nabíječka nabíjí jak OTG zařízení, tak i všechny připojené periférie [59]. Příkladem může být dokovací stanice LiveDock firmy Sony [60].

## 5.2 Bezdrátové nabíjení

OTG Y kabely s podporou napájení hostitelského zařízení jsou poměrně vzácné a podporují vždy jen vymezený okruh zařízení. Jejich použití tedy není univerzální. Bezdrátové nabíječky, které nepotřebují přístup k micro USB portu, který je využíván pro komunikaci, jsou oproti tomu na českém trhu dostupnější. Je však potřeba, aby dané mobilní zařízení podporovalo bezdrátové nabíjení.

Většina bezdrátových nabíječek funguje na principu elektromagnetické indukce mezi dvěma cívkami. V případě, že je podložka pro bezdrátové nabíjení připojena k nabíječce zapojené do sítě, vytváří cívka uvnitř ní elektromagnetické pole. Pokud je do tohoto pole vložena druhá cívka, která je umístěna v telefonu, indukuje cívka střídavý elektrický proud, který je usměrněn a následně přiveden do baterie. Účinnost bezdrátového nabíjení se odvíjí od použité technologie a materiálů a může dosahovat podobné účinnosti, jako mají klasické nabíječky [61].

Nejrozšířenějším standardem využívaným v bezdrátových nabíječkách je Qi, který vyvinulo Wireless Power Consortium, na kterém se podílí přes 220 firem (např. Samsung, Sony Ericsson, LG nebo HTC). Výhodou je, že všechny vysílače pracují se všemi přijímači. Všechna zařízení s podporou standardu Qi je tak možné napájet jedinou bezdrátovou nabíječkou [62].

Pro bezdrátové nabíjení mobilního telefonu Samsung Galaxy S4 jsem použila bezdrátovou nabíječku CONNECT IT CI-393, která je kompatibilní se standardem Qi [63]. Pro nabíjení je nutné vložit podložku pro bezdrátové nabíjení, určenou pro tento mobilní telefon, na kontakty umístěné pod krytem baterie telefonu [64]. Pro napájení je nutné mít telefon umístěn na podložce ve správné pozici a správném natočení. Nabíjení touto nabíječkou jsem testovala v kapitole 6.

# Kapitola 6

## Testování

### 6.0.1 Testování aplikace

Vytvořená aplikace byla testována na 3 různých zařízeních s OS Android:

**Tabulka 6.1:** Tabulka testovaných zařízení

	<b>IGET Blackview Zeta</b>	<b>Samsung Galaxy S4</b>	<b>Lenovo Yoga B8080</b>
Verze OS	4.4.2 (KitKat)	5.0.1 (Lollipop)	4.4.2 (KitKat)
API	19	21	19
Rozlišení displeje	1280x720 px	1920x1080 px	1920x1200 px
Velikost displeje	5"	5"	10.1"

UI aplikace se v orientaci na výšku i na šířku na všech zařízeních zobrazovalo správně a žádná jeho část nebyla překryta. Na všech zařízeních též byla testována funkčnost aplikace. Připojení ke zdroji bylo testováno pro následující případy:

- aplikace není spuštěna, zdroj je vypnut - aplikace se při připojení kabelu spustí a čeká, než se zdroj zapne, a následně se s ním spojí
- aplikace není spuštěna, zdroj je zapnut - aplikace se při připojení kabelu automaticky spustí a připojí ke zdroji
- aplikace je spuštěna, zdroj je vypnut - aplikace čeká, než se zdroj zapne, a následně se s ním spojí
- aplikace je spuštěna, zdroj je zapnut - při připojení kabelu se aplikace připojí ke zdroji

Zvláště v případě připojení k vypnutému zdroji aplikaci někdy trvalo déle, než se s ním spojila. V takovém případě bylo rychlejší kabel odpojit a znovu připojit, ačkoliv i bez toho ke spojení došlo.

Bylo též testováno připojení ke zdroji v různých částech aplikace, přičemž



byl testován případ, kdy se jednalo o prvotní připojení ke zdroji od spuštění aplikace, a případ, kdy byl kabel v průběhu používání aplikace odpojen a následně opět připojen. Funkčnost aplikace to na žádném testovaném zařízení neovlivnilo.

Editační pole byla správně svázána s posuvníky, při změně hodnoty došlo ke změně posuvníku a naopak při změně posuvníku došlo ke změně hodnoty zobrazené v editačním okně. Zároveň nebylo možné zadat do tohoto okna nepovolené znaky a hodnoty vyšší než maximální limit.

Spolehlivě též fungovalo vypínání a zapínání výstupu zdroje a ukládání aktuálně nastavených hodnot mezi profily.

Na všech zařízeních též fungovalo vytváření profilů a posílání jejich hodnot na zdroj. Seznam profilů v paměti zůstává i po vypnutí aplikace či telefonu, nikoliv však po jejím odinstalování, což ani není žádoucí.

Po aktualizování nastavení se toto nastavení aktualizuje v celé aplikaci.

Vytváření, upravování i mazání programů fungovalo na všech zařízeních spolehlivě. Jediným nedostatkem zde bylo to, že při úpravě první položky nedošlo k automatickému doplnění hodnot do editačních polí.

Na všech zařízeních též fungoval běh programu, který byl testován i se zátěží připojenou ke zdroji, která při napětí 17 V odebírala proud 8 A. Při odpojení od zdroje v průběhu běhu programu byl program pozastaven a po opětovném připojení mohl dále pokračovat stisknutím zeleného tlačítka "Start". Správně fungovalo i úplné zastavení programu a jeho následné spuštění od začátku, přičemž po jeho novém spuštění se začalo nové zaznamenávání průběhu hodnot napětí, proudu a elektrického náboje v čase. Program zůstal běžet i při vypnutí obrazovky, otevření jiné aplikace nebo zobrazení průběhu programu v grafu či textové podobě.

Hodnoty průběhu programu v grafu a seznamu byly aktualizovány s během programu, bylo tak možné sledovat samotný graf při spuštěném programu. Ukládání průběhu programu v grafu jako obrázku a ve formátu CSV fungovalo spolehlivě jak na zařízeních s vloženou paměťovou kartou, tak i na zařízeních bez ní.

Též posílání e-mailů fungovalo správně jak při připojení k síti Wi-Fi, tak i při připojení k mobilní síti. Pokud zařízení nemělo přístup k internetu vůbec, zobrazila se ihned hláška, že e-mail nemohl být poslán.

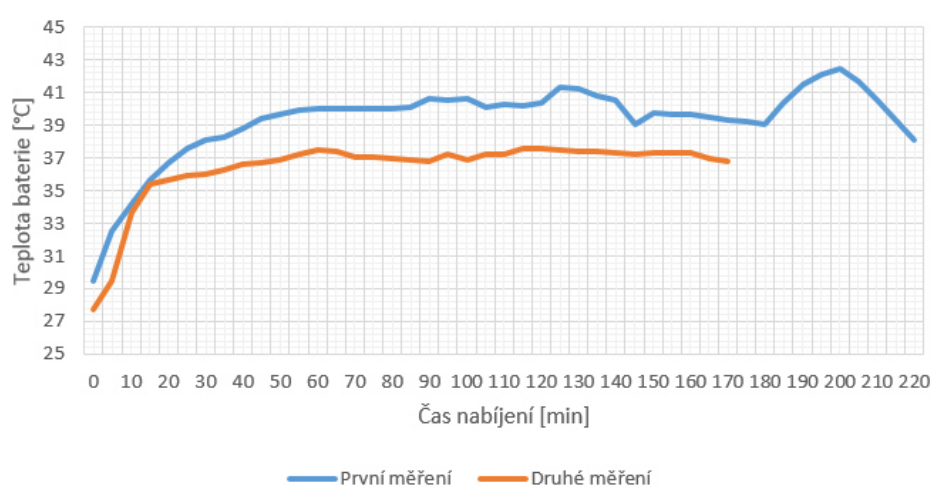
Detekci výpadku síťového napájení jsem testovala s použitím bezdrátové nabíječky pouze na zařízení Samsung Galaxy S4, jelikož jako jediné z testovaných zařízení podporuje standard Qi. Detekce napájení fungovala a při výpadku napájení byly nastaveny definované hodnoty napětí a proudu na zdroj a poslán e-mail uživateli v případě, že byla zaškrtnuta příslušná pole v nastavení.

## 6.0.2 Testování bezdrátové nabíječky

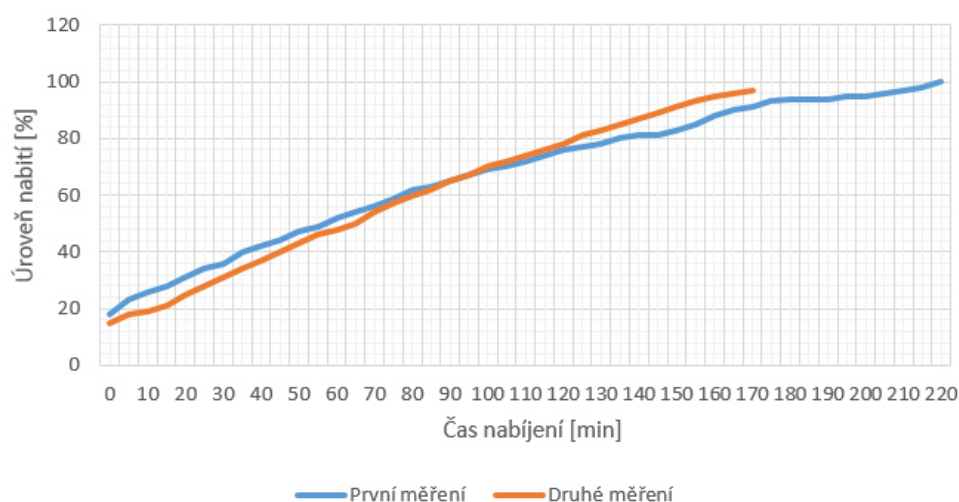
Při nabíjení dochází k zahřívání baterie, proto jsem zkoumala její zahřívání v závislosti na čase. Tuto závislost je možné vidět v grafu na obrázku 6.1. Při prvním měření, které je v grafu znázorněno modře, byla počáteční úroveň

baterie 18% a naplno se nabila za 3 hodiny a 40 minut. Při druhém měření, které je v grafu znázorněno oranžovou barvou, byla její počáteční úroveň 15% a její napájení skončilo v 97% úrovni baterie. Druhé nabíjení trvalo kratší dobu, 2 hodiny a 50 minut. Z grafu je patrné, že největší nárůst teploty se děje v první půlhodině nabíjení. Poté se teplota baterie již téměř nemění. U prvního měření si lze povšimnout většího kolísání teplot, které může být způsobeno větší manipulací se zařízením. U druhého měření jsem se snažila s telefonem manipulovat minimálně. Při nabíjení ani u jednoho z případů nedošlo k přerušení nabíjení telefonu. Maximální zaznamenaná hodnota teploty baterie byla 42,5°C.

Závislost úrovně nabití na době nabíjení je možné vidět na obrázku 6.2. Tato závislost je téměř lineární. Za půl hodiny bylo u prvního měření nabití průměrně přes 10% baterie, u druhého nabíjení pak necelých 15%.



**Obrázek 6.1:** Graf závislosti teploty baterie na čase nabíjení



**Obrázek 6.2:** Graf závislosti úrovně nabití baterie na čase nabíjení

# Kapitola 7

## Závěr

Cílem této bakalářské práce bylo vytvořit mobilní aplikaci pro ovládání programovatelného laboratorního zdroje Manson HCS-3604-USB spolu s funkcemi vyčtení a nastavení hodnot na zdroji, správy profilů a programů, včetně výpočtu elektrického náboje, a detekce výpadku síťového napájení.

Tento zdroj, společně s oficiální aplikací pro jeho ovládání, je popsán v první kapitole této práce. V druhé kapitole jsem se pak zabývala operačním systémem Android a základními informacemi pro vývoj aplikací na této platformě. V praktické části je pak popsána vytvořená aplikace včetně napájení telefonu při současné komunikaci se zdrojem a jejího testování.

Pro ovládání zdroje existuje oficiální aplikace, ke které je ovšem nutné dokoupit Raspberry Pi a nainstalovat potřebný driver. Navíc aplikace neumožňuje některé funkce, například detekci výpadku síťového napájení, a z jejího testování vyplynulo, že není zcela spolehlivá.

Vytvořená aplikace oproti tomu využívá pro komunikaci snadno dostupný OTG kabel. Navíc podporuje více zařízení s OS Android, jelikož ji lze používat i na zařízeních s menší velikostí obrazovky. Aplikace má implementovány všechny výše zmíněné funkce a dále také funkce zobrazení průběhu programu v grafu a textové podobě, včetně jejich uložení do souboru, a posílání zpráv na e-mail. Nejjednodušším způsobem napájení telefonu při současné komunikaci se zdrojem je nabíjení pomocí bezdrátové nabíječky.



## Literatura

- [1] HCS-3600/3602/3604 USB: Laboratory Grade & High RFI Immunity Switching Mode Power Supply with Rotary Encoder Control - User Manual. *Manson Engineering Industrial Ltd* [online]. 2015 [cit. 2017-05-08] Dostupné z: [www.manson.com.hk/getimage/index/action/images/name/5652d716c3b11.pdf](http://www.manson.com.hk/getimage/index/action/images/name/5652d716c3b11.pdf)
- [2] Constant voltage power supply and constant current power supply. *Manson Engineering Industrial Ltd* [online]. [cit. 2017-05-10] Dostupné z: <http://www.manson.com.hk/getfile/index/action/product/name/53f4654a6f976.pdf>
- [3] Aplikace pro Android ve službě GooglePlay - *Manson Power Supply remote* [online]. [cit. 2017-05-08] Dostupné z: <https://play.google.com/store/apps/details?id=com.yzd.manson>
- [4] *Manson Engineering Industrial Ltd* [online]. 2013 [cit. 2017-05-08] Dostupné z: <http://manson.com.hk/>
- [5] Raspberry Pi driver installation steps. *Manson Engineering Industrial Ltd* [online]. 2015 [cit. 2017-05-08] Dostupné z: [http://www.manson.com.hk/getimage/index/action/images/name/Raspberry\\_pi\\_driver\\_install\\_1435303994.pdf](http://www.manson.com.hk/getimage/index/action/images/name/Raspberry_pi_driver_install_1435303994.pdf)
- [6] Global mobile OS market share in sales to end users from 1st quarter 2009 to 1st quarter 2016. *Statista – The portal for statistics* [online]. [cit. 2017-05-11]. Dostupné z: <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>
- [7] Market share of mobile operating systems in Czech Republic from 2010 to 2016. *Statista – The portal for statistics* [online]. [cit. 2017-05-11]. Dostupné z: <https://www.statista.com/statistics/669587/market-share-mobile-operating-systems-czech-republic/>
- [8] LACKO, Euboslav. *Vývoj aplikací pro Android*. Brno: Computer Press, 2015. ISBN 978-80-251-4347-6.

- [9] Codenames, Tags, and Build Numbers. *Android Open Source Project* [online]. [cit. 2017-05-12]. Dostupné z: <https://source.android.com/source/build-numbers>
- [10] Build.VERSION\_CODES. *Android Developers* [online]. [cit. 2017-05-12]. Dostupné z: [https://developer.android.com/reference/android/os/Build.VERSION\\_CODES.html](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html)
- [11] Android History: The Android Story. *Android* [online]. [cit. 2017-05-12]. Dostupné z: <https://www.android.com/history/>
- [12] Android Version History: A Timeline of New Features. *Tom's Guide* [online]. 2015 [cit. 2017-05-12]. Dostupné z: <http://www.tomsguide.com/us/android-version-history,news-21211.html>
- [13] Android: Marshmallow. *Android* [online]. 2017 [cit. 2017-05-12]. Dostupné z: <https://www.android.com/versions/marshmallow-6-0/>
- [14] Android: Nougat. *Android* [online]. 2017 [cit. 2017-05-12]. Dostupné z: <https://www.android.com/versions/nougat-7-0/>
- [15] Android Distribution Updated for May 2017: Nougat Rockets to Over 7%. *Droid Life* [online]. 2017 [cit. 2017-05-12]. Dostupné z: <http://www.droid-life.com/2017/05/02/distribution-may-2017/>
- [16] Application Fundamentals. *Android Developers* [online]. [cit. 2017-05-12]. Dostupné z: <https://developer.android.com/guide/components/fundamentals.html>
- [17] Activity. *Android Developers* [online]. [cit. 2017-05-12]. Dostupné z: <https://developer.android.com/reference/android/app/Activity.html>
- [18] App Manifest. *Android Developers* [online]. [cit. 2017-05-13]. Dostupné z: <https://developer.android.com/guide/topics/manifest/manifest-intro.html>
- [19] Configure Your Build. *Android Developers* [online]. [cit. 2017-05-13]. Dostupné z: <https://developer.android.com/studio/build/index.html>
- [20] Fragments. *Android Developers* [online]. [cit. 2017-05-13]. Dostupné z: <https://developer.android.com/guide/components/fragments.html>
- [21] Intents and Intent Filters. *Android Developers* [online]. [cit. 2017-05-17]. Dostupné z: <https://developer.android.com/guide/components/intents-filters.html>
- [22] Supporting Multiple Screens. *Android Developers* [online]. [cit. 2017-05-15]. Dostupné z: [https://developer.android.com/guide/practices/screens\\_support.html](https://developer.android.com/guide/practices/screens_support.html)

- [23] Accessing Resources. *Android Developers* [online]. [cit. 2017-05-15]. Dostupné z: <https://developer.android.com/guide/topics/resources/accessing-resources.html>
- [24] UI Overview. *Android Developers* [online]. [cit. 2017-05-15]. Dostupné z: <https://developer.android.com/guide/topics/ui/overview.html>
- [25] Input Controls. *Android Developers* [online]. [cit. 2017-05-15]. Dostupné z: <https://developer.android.com/guide/topics/ui/controls.html>
- [26] DatePicker. *Android Developers* [online]. [cit. 2017-05-16]. Dostupné z: <https://developer.android.com/reference/android/widget/DatePicker.html>
- [27] NavigationView. *Android Developers* [online]. [cit. 2017-05-16]. Dostupné z: <https://developer.android.com/reference/android/support/design/widget/NavigationView.html>
- [28] ScrollView. *Android Developers* [online]. [cit. 2017-05-16]. Dostupné z: <https://developer.android.com/reference/android/widget/ScrollView.html>
- [29] Layouts. *Android Developers* [online]. [cit. 2017-05-16]. Dostupné z: <https://developer.android.com/guide/topics/ui/declaring-layout.html#AdapterViews>
- [30] Grid View. *Android Developers* [online]. [cit. 2017-05-16]. Dostupné z: <https://developer.android.com/guide/topics/ui/layout/gridview.html>
- [31] List View. *Android Developers* [online]. [cit. 2017-05-16]. Dostupné z: <https://developer.android.com/guide/topics/ui/layout/listview.html>
- [32] Spinners. *Android Developers* [online]. [cit. 2017-05-16]. Dostupné z: <https://developer.android.com/guide/topics/ui/controls/spinner.html>
- [33] Structure. *Material Design Guidelines: Layout* [online]. [cit. 2017-05-16]. Dostupné z: <https://material.io/guidelines/layout/structure.html>
- [34] Material Design for Developers. *Android Developers* [online]. [cit. 2017-05-16]. Dostupné z: <https://developer.android.com/training/material/index.html>
- [35] Environment. *Material Design Guidelines: Material Design* [online]. [cit. 2017-05-16]. Dostupné z: <https://material.io/guidelines/material-design/environment.html#environment-light-shadow>

- [36] Material Properties. *Material Design Guidelines: Material Design* [online]. [cit. 2017-05-16]. Dostupné z: <https://material.io/guidelines/material-design/material-properties.html>
- [37] Elevation & Shadows. *Material Design Guidelines: Material Design* [online]. [cit. 2017-05-16]. Dostupné z: <https://material.io/guidelines/material-design/elevation-shadows.html>
- [38] Color. *Material Design Guidelines: Style* [online]. [cit. 2017-05-16]. Dostupné z: <https://material.io/guidelines/style/color.html>
- [39] Buttons. *Material Design Guidelines: Components* [online]. [cit. 2017-05-16]. Dostupné z: <https://material.io/guidelines/components/buttons.html>
- [40] Buttons: Floating Action Button. *Material Design Guidelines: Components* [online]. [cit. 2017-05-16]. Dostupné z: <https://material.io/guidelines/components/buttons-floating-action-button.html>
- [41] Navigation Drawer. *Material Design Guidelines: Patterns* [online]. [cit. 2017-05-16]. Dostupné z: <https://material.io/guidelines/patterns/navigation-drawer.html>
- [42] Dialogs. *Material Design Guidelines: Components* [online]. [cit. 2017-05-16]. Dostupné z: <https://material.io/guidelines/components/dialogs.html>
- [43] Kyleduo/SwitchButton: A cute widget of Switch Button for you to create beautiful and friendly UI. *GitHub: The world's leading software development platform* [online]. [cit. 2017-05-22]. Dostupné z: <https://github.com/kyleduo/SwitchButton>
- [44] Appsthatmatter/GraphView: Android Graph Library for creating zoomable and scrollable line and bar graphs. *GitHub: The world's leading software development platform* [online]. [cit. 2017-05-22]. Dostupné z: <https://github.com/appsthatmatter/GraphView>
- [45] Basics of GraphView. *Android Graph View plotting library* [online]. 2017 [cit. 2017-05-19]. Dostupné z: [www.android-graphview.org/basics-of-graphview/](http://www.android-graphview.org/basics-of-graphview/)
- [46] Style options. *Android Graph View plotting library* [online]. 2017 [cit. 2017-05-19]. Dostupné z: <http://www.android-graphview.org/style-options/>
- [47] Realtime chart. *Android Graph View plotting library* [online]. 2017 [cit. 2017-05-19]. Dostupné z: <http://www.android-graphview.org/realtime-chart/>
- [48] FelHR85/UsbSerial: Usb serial controller for Android. *GitHub: The world's leading software development platform* [online]. [cit. 2017-05-22]. Dostupné z: <https://github.com/felHR85/UsbSerial>

- [49] UsbSerial: A serial port driver library for Android v4.5. *Felhr85* [online]. 2014 [cit. 2017-05-20]. Dostupné z: <https://felhr85.net/2014/11/11/usbserial-a-serial-port-driver-library-for-android-v2-0/>
- [50] USB Host. *Android Developers* [online]. [cit. 2017-05-20]. Dostupné z: <https://developer.android.com/guide/topics/connectivity/usb/host.html>
- [51] Saving Files. *Android Developers* [online]. [cit. 2017-05-20]. Dostupné z: <https://developer.android.com/training/basics/data-storage/files.html>
- [52] Javamail-android. *Google Code Archive: Long-term storage for Google Code Project Hosting* [online]. [cit. 2017-05-22]. Dostupné z: <https://code.google.com/archive/p/javamail-android/>
- [53] JavaMail API: Core Classes. *Tutorials Point* [online]. [cit. 2017-05-21]. Dostupné z: [https://www.tutorialspoint.com/javamail\\_api/javamail\\_api\\_core\\_classes.htm](https://www.tutorialspoint.com/javamail_api/javamail_api_core_classes.htm)
- [54] On-The-Go and Embedded Host Supplement to the USB Revision 2.0 Specification. *Universal Serial Bus* [online]. 2009 [cit. 2017-05-23]. Dostupné z: [http://www.usb.org/developers/onthego/USB\\_OTG\\_and\\_EH\\_2-0.pdf](http://www.usb.org/developers/onthego/USB_OTG_and_EH_2-0.pdf)
- [55] USB On-the-Go Basics. *Maxim Integrated: Analog, linear, & mixed-signal devices* [online]. [cit. 2017-05-23]. Dostupné z: <https://www.maximintegrated.com/en/app-notes/index.mvp/id/1822>
- [56] VOJÁČEK, Antonín. Co se skrývá pod komunikací označenou jako USB OTG? *Vývoj.HW.cz: Vše o elektronice a programování* [online]. 2008 [cit. 2017-05-23]. Dostupné z: <http://vyvoj.hw.cz/teorie-a-praxe/dokumentace/co-se-skrывa-pod-komunikaci-oznacenu-jako-usb-otg.html>
- [57] JOYCE, John. On-the-Go! Fascinating Facts about USB OTG. *Scientific Computing: Informatics, HPC & IT Solutions* [online]. 2015 [cit. 2017-05-23]. Dostupné z: <https://www.scientificcomputing.com/blog/2015/01/go-fascinating-facts-about-usb-otg>
- [58] Industrial IoT: Micro USB Host OTG Y-Cable with Micro USB Power Charging for Samsung Phones. *Valarm: Industrial IoT, Remote Monitoring, Mobile Sensor Telemetry* [online]. 2015 [cit. 2017-05-23]. Dostupné z: <https://shop.valarm.net/collections/cables-connectors/products/micro-usb-host-otg-y-cable-with-micro-usb-power-for-samsung>
- [59] Battery Charging Specification. *USB Implementers Forum, Inc.* [online]. 2010 [cit. 2017-05-23]. Dostupné z: [http://composter.com.ua/documents/BC1.2\\_FINAL.pdf](http://composter.com.ua/documents/BC1.2_FINAL.pdf)



- [60] Sony DK10 LiveDock Multimedia Station. *Amazon.co.uk: Low Prices in Electronics, Books, Sports Equipment & more* [online]. [cit. 2017-05-23]. Dostupné z: <https://www.amazon.co.uk/Sony-DK10-LiveDock-Multimedia-Station/dp/B005HPY5U0>
- [61] BOČEK, Vít. Jak funguje bezdrátové nabíjení (nejen) u Lumií. *MobilMania.cz: O mobilech víme vše* [online]. 2012 [cit. 2017-05-23]. Dostupné z: <http://www.mobilmania.cz/clanky/jak-funguje-bezdratove-nabijeni-nejen-u-lumii/sc-3-a-1321953/>
- [62] Wireless Charging Technology: ZENS Wireless Charging. *ZENS: Wireless Charging Solutions* [online]. [cit. 2017-05-23]. Dostupné z: <https://www.makezens.com/wireless-charging-technology/>
- [63] Bezdrátová nabíječka. *Connect IT* [online]. [cit. 2017-05-23]. Dostupné z: <http://www.connectit-europe.com/produkt/bezdratova-nabijecka/>
- [64] Podložka pro bezdrátové nabíjení pro Samsung Galaxy S4. *Connect IT* [online]. [cit. 2017-05-23]. Dostupné z: <http://www.connectit-europe.com/produkt/podlozka-pro-bezdratove-nabijeni-pro-samsung-galaxy-s4/>

## ■ 7.1 Seznam zkratek

<b>ADP</b> .....	Attach Detection Protocol
<b>API</b> .....	Application Programming Interface (rozhraní pro programování aplikací)
<b>CC</b> .....	Contant Current (konstantní proud)
<b>CV</b> .....	Constant Voltage (konstantní napětí)
<b>dp</b> .....	Density Independent Pixel (pixel nezávislý na denzitě)
<b>DPI</b> .....	Dots Per Inch (bodů na palec)
<b>GUI</b> .....	Graphical User Interface (grafické uživatelské rozhraní)
<b>HNP</b> .....	Host Negotiation Protocol
<b>NFC</b> .....	Near Field Communication (komunikace v blízkém poli)
<b>OS</b> .....	Operating System (operační systém)
<b>OTG</b> .....	On-The-Go
<b>SMTP</b> .....	Simple Mail Transfer Protocol
<b>SRP</b> .....	Session Request Protocol
<b>TPL</b> .....	Targeted Peripheral List
<b>UI</b> .....	User Interface (uživatelské rozhraní)
<b>XML</b> .....	Extensible Markup Language (rozšiřitelný značkovací jazyk)





## Přílohy

# Příloha A

## Tabulky

**Tabulka A.1:** Příkazy pro komunikaci se zdrojem [1]

Vstupní příkaz a vrácená hodnota	Funkce	Příklad (Příkaz » Odpověď, Význam)
GMAX[CR]  <napětí><proud>[CR] OK[CR]	Maximální hodnota napětí a proudu zdroje  <napětí>=xxx <proud>=xxx	GMAX[CR] » 180200[CR] OK[CR]  Maximální napětí je 18.0V Maximální proud je 20.0A
SOUT<stav>[CR]  OK[CR]	Zapne/vypne výstup zdroje  <stav>=0/1 (0=zapnutí, 1 = vypnutí)	SOUT0[CR] » OK[CR]  Zapnutí výstupu zdroje
VOLT<napětí>[CR]  OK[CR]	Nastaví hodnotu napětí  <napětí>=000 < xxx < Max-Volt	Volt127[CR] » OK[CR]  Nastaví hodnotu napětí na 12.7V
CURR<proud>[CR]  OK[CR]	Nastaví hodnotu proudu  <proud>=000 < xxx < Max-Curr	CURR120[CR] » OK[CR]  Nastaví hodnotu proudu na 12.0A

<p>GETS[CR]</p> <p>&lt;napětí&gt;&lt;proud&gt;[CR]</p> <p>OK[CR]</p>	<p>Získá nastavenou hodnotu napětí a proudu</p> <p>&lt;napětí&gt;=xxx</p> <p>&lt;proud&gt;=xxx</p>	<p>GETS[CR]</p> <p>»</p> <p>150110[CR]</p> <p>OK[CR]</p> <p>Hodnota napětí je nastavena na 15V</p> <p>Hodnota proudu je nastavena na 11A</p>
<p>GETD[CR]</p> <p>&lt;napětí&gt;&lt;proud&gt;</p> <p>&lt;stav&gt;[CR]</p> <p>OK[CR]</p>	<p>Získá hodnoty napětí, proudu a stavu CC/CV na displeji</p> <p>&lt;napětí&gt;=xxxx</p> <p>&lt;proud&gt;=xxxx</p> <p>&lt;stav&gt;=0/1 (0=CV, 1=CC)</p>	<p>GETD[CR]</p> <p>»</p> <p>150016001[CR]</p> <p>OK[CR]</p> <p>Hodnoty zobrazené na displeji jsou 15V a 16A</p> <p>Zdroj je v módu CC</p>
<p>GOVP[CR]</p> <p>&lt;napětí&gt;[CR]</p> <p>OK[CR]</p>	<p>Získá nastavený horní limit výstupního napětí</p> <p>&lt;napětí&gt;=xxx</p>	<p>GOVP[CR]</p> <p>»</p> <p>251[CR]</p> <p>OK[CR]</p> <p>Nastavený horní limit výstupního napětí je 25.1V</p>
<p>SOVP&lt;napětí&gt;[CR]</p> <p>OK[CR]</p>	<p>Nastaví horní limit výstupního napětí</p> <p>&lt;napětí&gt;=000 &lt; xxx &lt; Max-Volt</p>	<p>SOVP151[CR]</p> <p>»</p> <p>OK[CR]</p> <p>Nastaví horní limit výstupního napětí na 15.1V</p>
<p>GOCP[CR]</p> <p>&lt;proud&gt;[CR]</p> <p>OK[CR]</p>	<p>Získá nastavený horní limit výstupního proudu</p> <p>&lt;proud&gt;=xxx</p>	<p>GOCP[CR]</p> <p>»</p> <p>091[CR]</p> <p>OK[CR]</p> <p>Nastavený horní limit výstupního proudu je 9.1A</p>

<p>SOCP&lt;proud&gt;[CR] OK[CR]</p>	<p>Nastaví horní limit výstupního proudu  &lt;proud&gt;=000 &lt; xxx &lt; Max-Curr</p>	<p>SOCP151[CR] » OK[CR]  Nastaví horní limit výstupního proudu na 15.1A</p>
<p>PROM &lt;napětí0&gt;&lt;proud0&gt; &lt;napětí1&gt;&lt;proud1&gt; &lt;napětí2&gt;&lt;proud2&gt;[CR] OK[CR]</p>	<p>Uloží hodnotu napětí a proudu do 3 míst v paměti zdroje  &lt;napětíX&gt;=xxx &lt;currentX&gt;=xxx (X je číslo místa paměti od 0 do 2)</p>	<p>PROM111111022122033133 [CR] » OK[CR]  Nastaví paměť 0 na 11.1V a 11.1A Nastaví paměť 1 na 2.2V a 12.2A Nastaví paměť 2 na 3.3V a 13.3A</p>
<p>GETM[CR]  &lt;napětí0&gt;&lt;proud0&gt;[CR] &lt;napětí1&gt;&lt;proud1&gt;[CR] &lt;napětí2&gt;&lt;proud2&gt;[CR] OK[CR]</p>	<p>Získá uložené hodnoty napětí a proudu ze 3 míst v paměti zdroje  &lt;napětíX&gt;=xxx &lt;proudX&gt;=xxx (X je číslo místa paměti od 0 do 2 )</p>	<p>GETM[CR] » 111111[CR] 122122[CR] 144145[CR] OK[CR]  V paměti 0 jsou uloženy hodnoty 11.1V a 11.1A V paměti 1 jsou uloženy hodnoty 12.2V a 12.2A V paměti 2 jsou uloženy hodnoty 14.4V a 14.5A</p>
<p>RUNM&lt;paměť&gt;[CR] OK[CR]</p>	<p>Nastaví napětí a proud podle hodnot uvnitř dané paměti zdroje  &lt;paměť&gt;=0/1/2</p>	<p>RUNM1[CR] » OK[CR]  Nastaví napětí a proud na hodnoty uložené v paměti 1</p>

**Tabulka A.2:** Přehled verzí OS Android [9] [10]

Číselné označení	Kódové označení	Verze API	Měsíc a rok vydání
1.0	žádné	1	říjen 2008
1.1	žádné	2	únor 2009
1.5	Cupcake	3	květen 2009
1.6	Donut	4	září 2009
2.0	Eclair	5	listopad 2009
2.0.1	Eclair	6	prosinec 2009
2.1	Eclair	7	leden 2010
2.2	Froyo	8	červen 2010
2.3 - 2.3.2	Gingerbread	9	listopad 2010
2.3.3 - 2.3.7	Gingerbread	10	únor 2011
3.0	Honeycomb	11	únor 2011
3.1	Honeycomb	12	květen 2011
3.2	Honeycomb	13	červen 2011
4.0.1 - 4.0.2	Ice Cream Sandwich	14	říjen 2011
4.0.3 - 4.0.4	Ice Cream Sandwich	15	prosinec 2011
4.1	Jelly Bean	16	červen 2012
4.2	Jelly Bean	17	listopad 2012
4.3	Jelly Bean	18	červenec 2013
4.4 - 4.4.4	KitKat	19	říjen 2013
4.4W	KitKat Watch	20	červen 2014
5.0	Lollipop	21	listopad 2014
5.1	Lollipop	22	březen 2015
6.0	Marshmallow	23	říjen 2015
7.0	Nougat	24	srpen 2016
7.1	Nougat	25	říjen 2016