TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Multifunkční panel pro elektromobil

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802T007 – Informační technologie

Autor práce: **Tomáš Procházka**
Vedoucí práce: Ing. Přemysl Svoboda

Liberec 2016

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

Multifunction panel for electric car

Bachelor thesis

Study programme: B2646 – Information technology
Study branch: 1802T007 – Information technology

Author: **Tomáš Procházka**
Supervisor: Ing. Přemysl Svoboda

Liberec 2016

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: **2015/2016**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Tomáš Procházka**
Osobní číslo:	**M11000112**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Multifunkční panel pro elektromobil**
Zadávající katedra:	**Ústav mechatroniky a technické informatiky**

Zásady pro vypracování:

1. Seznamte se s platformou s OS Android, programováním pro tento systém v jazyce Java a možností přizpůsobení zařízení s OS Android pro kioskový režim.
2. Seznamte se s možnostmi integrace tabletu jako multifunkčního panelu do elektromobilu eŠus vyvíjeného na TUL, případně navrhněte způsob ovládání funkcí elektromobilu (audio systém, osvětlení, přístup k internetu, čtení informací o palubní síti a další).
3. Naprogramujte aplikaci pro multifunkční panel, která bude realizovat ovládání alespoň dvou funkcí elektromobilu (ad. bod 2).

Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah pracovní zprávy: **cca 30–40 stran**

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

[1] **Programování v jazyce Java, David Flanagan, Praha : Computer Press, 1997**

[2] **Android 2, Mark L. Murphy, Praha: Computer Press, 2011**

Vedoucí bakalářské práce: **Ing. Přemysl Svoboda**
Ústav mechatroniky a technické informatiky

Konzultant bakalářské práce: **Ing. Pavel Jandura**
Ústav mechatroniky a technické informatiky

Datum zadání bakalářské práce: **10. října 2015**

Termín odevzdání bakalářské práce: **16. května 2016**

L.S.

prof. Ing. Václav Kopecký, CSc.
děkan

doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2015

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 11. 5. 2016

Podpis: Pavlína Zlá

Abstrakt

Multifunkčním panelem pro elektromobil je myšleno zařízení, konkrétně tablet, zabudovaný do palubní desky školního elektromobilu eŠus vyvíjeného na Technické univerzitě v Liberci.

Cílem je využít tablet jako panelový displej pro školní elektromobil. Tablet byl zakoupen s operačním systémem Android a pro tento operační systém byla vytvořena aplikace poskytující kioskový režim. Tato aplikace nabízí další vytvořené funkce, avšak využitelné pouze ve školním elektromobilu. Těmito funkcemi je například myšleno ovládání audio systému, GPS navigace, detekce osvětlení, čtení informací o palubní síti elektromobilu a další.

V současné době jsou již vytvořené následujcí aplikace Zavaděč, MP3 přehrávač, Osvětlení a Informační panel. Zavaděč představuje aplikaci s kioskovým režimem, je spuštěný hned po zapnutí tabletu a jím lze spustit další konkrétní aplikace vytvořené k využití ve školním elektromobilu. Jednou z těchto aplikací je aplikace MP3 přehrávač, sloužící k běžnému přehrávání mp3 souborů v tabletu. Další spustitelnou aplikací ze Zavaděče je Osvětlení, které graficky znázornuje rozsvícení a zhasnutí světel elektromobilu, tzn. typických světelných prvků vozidla na pozemní komunikaci. Pro simulační účely byly vytvořeny ovládací prvky k ovládání světel elektromobilu, přičemž se využívá externího hardwaru UMFT311EV. Poslední možnou aplikací, již lze Zavaděčem spustit, je aplikace Informační panel v podobě palubní desky elektromobilu, který je schopný zpracovat informační rámec ze sběrnice CAN. Získaný rámec zpracuje a posléze promítne informace v grafickém provedení na displej tabletu. Dále byl proveden root konkrétně použitého tabletu, díky němuž byly odblokovány rozšířené možnosti, které budou využity v dalším vývoji a rozvoji nových aplikací.

Aplikace spojené s činností a budoucím vývojem elektromobilu jsou cílem mé práce. Je možné je využít pro další potřeby a možnosti elektromobilu vzhledem k jeho pokračujícímu vývoji.

Klíčová slova

Android, elektromobil eŠus, kioskový režim, multifunkční panel, sběrnice CAN

Abstract

An electric car's multifunctional panel is a device, specifically a tablet, installed into the dashboard of the school electric car called eŠus developed at the Technical University of Liberec.

The aim is to use a tablet as a panel display of the school electric car. The tablet that was bought has the Android operation system and an application providing the kiosk mode for this system was created. The application provides further created functions usable for the car, such as audio system control, GPS navigation, lights detection, reading information about the car's on-board electrical system etc.

So far, the applications Launcher, MP3 Player, Lights and Information Panel have been created. Launcher, the application with the kiosk mode, is turned on after the tablet does and can start other applications usable in the car. These applications are MP3 Player (for playing mp3 files in the tablet) and Lights (graphically indicates turning the lights in the car – required to drive the car on the roads – on and off). Control components were created to simulate the lights control using external hardware UMFT311EV. The last application startable by the Launcher is Information Panel resembling the electric car's dashboard. It can process frame with the information from the CAN bus. The gained information from this frame are processed and graphically represented on the tablet's display. Also, the used tablet was rooted, which opened wide variety of options usable in the development of new applications.

The applications connected with the operating and future development of the electric car are the aim of my work. They are usable for further needs and options of the car, considering its progressive development.

Keywords

Android, electric car eŠus, kiosk mode, multifunctional panel, CAN bus

Poděkování

Rád bych poděkoval vedoucímu práce panu Ing. Přemyslu Svobodovi za sestavení zadání mé bakalářské práce, rovněž za půjčení FTDI zařízení a podporu při konzultačních hodinách.

Chci poděkovat i mému konzultantovi panu Ing. Pavlu Jandurovi za umožnění téma mé bakalářské práce, sehnání a zapůjčení tabletu ASUS MeMO Pad 8, na němž byla práce realizována.

Obsah

Seznam obrázků

Seznam zdrojových kódů

Seznam zkratek

CAN	Controller Area Network, sběrnice využívaná pro automobilovou diagnostiku
FTDI	Future Technology Devices International, Mezinárodní budoucí technologická zařízení (název firmy)
FM	Fakulta mechatroniky, informatiky a mezioborových studií Technické univerzity v Liberci
Full HD	vysoké rozlišení
GPIO	General-purpose input/output, vstupy a výstupy programovatelné softwarem
GPS	Global Positioning System, globální polohovací systém
JDK	Java Development Kit, nástroje pro vývoj aplikací pro platformu Java
Linux	volně šiřitelný operační systém
LTE	Long Term Evolution, vysokorychlostní internet v mobilních sítích
OS	operační systém
RAM	Random Access Memory, polovodičová paměť s možností čtení a zápisu
SOC	state of charge, stav nabití
TUL	Technická univerzita v Liberci
UI	User Interface, uživatelské rozhraní
USB	Universal Serial Bus, univerzální sériová sběrnice
XML	Extensible Markup Language, značkovací jazyk

1 Úvod

Vývoj technologií jde neustále kupředu. Dnes lidé ve velké míře používají tablet, zařízení, jenž je ovládáno pomocí dotykového displeje. K nejčastějším operační systémům v tabletu patří OS Android. Otevřenost tohoto systému nám dává možnost vytvářet různé aplikace specializované na odlišné činnosti a tyto aplikace jsou následně zdarma poskytovány lidem prostřednictvím internetového obchodu Google Play, který vznikl právě z tohoto důvodu. Díky velkému množství již vytvořených aplikací, lze tablet velmi snadno přizpůsobit potřebným představám.

1.1 Využití tabletu

Tablet využívají například děti ve školách. Jsou jim na nich nabízeny různé vzdělávací aplikace a výukové programy. Ty modernizují výuku nejen v základní škole, ale také na středních školách. Dokonce ve vysokoškolském prostředí jsou tablety vhodným prostředkem pro vzájemnou komunikaci se spolužáky či profesory, ať už pomocí chatu, emailu nebo jiných komunikačních metod. Možnost připojit se k internetu dělá z tabletu výborný zdroj k získávání informací. Dokáže tedy nahradit učebnice, sešit s poznámkami a poskytne spoustu dalších možností.

V posledních letech je dotykový displej využíván v oblasti automobilů. S jeho pomocí lze spoustu běžných problémů snadno vyřešit. Poskytne například poslech hudby prostřednictvím internetových rádií nebo hudebních souborů v připojeném mobilním telefonu. Zvládne nahradit mapu díky aplikaci s mapovým materiálem a GPS navigací, která zobrazí aktuální polohu a bez velkých komplikací nás dovede do cíle. Poskytuje filtrování trasy, které zahrnuje neplacené úseky, nejkratší cestu, nejrychlejší cestu i aktuální dopravní situaci. Na displeji je možné kontrolovat stav vozidla, naleznout případnou závadu, zjistit opotřebení některých součástí vozidla a varuje nás ještě před jejich poškozením. Dále díky kamerovému systému, čidlům a dalším senzorům lze zobrazit vzdálenost okolních předmětů od vozidla, a proto umožňuje snadnější zaparkování. Také existuje možnost automatického parkovacího pilota, který zaparkuje zcela bez zásahu uživatele. Internet v automobilu umožní najít cíle výletů, blízké restaurace a poskytne aktuální informace o dopravní situaci.

1.2 Tablet v eŠusu

Na Technické univerzitě v Liberci je stále vyvíjen školní elektromobil s názvem eŠus (obr. 1.1) a nyní probíhá výzkum o využití tabletu jako multifunkčního panelu. Zjišťují se možnosti použití tabletu jako snadno dostupné alternativy k běžným, drahým automobilovým systémům, neboť též poskytuje dotykovou obrazovku. Tablet díky operačnímu systému Android nabízí možnosti pro vytvoření potřebných aplikací. Tablet bude umístěný uprostřed palubní desky, tedy po pravé straně řidiče pro snadné ovládání při řízení vozidla. Pro snadné ovládání budou vytvořeny aplikace v jednoduchém provedení a dostatečnou velikostí ovládacích prvků.

Obrázek 1.1: elektromobil eŠus

Tablet poskytne kioskový režim v podobě aplikace, která umožní snadné ovládání s omezeným počtem aplikací. Mezi tyto aplikace patří aplikace pro potřeby elektromobilu a po spuštění tabletu budou k dispozici. Aplikace kioskového režimu bude obsahovat úvodní obrazovku s hodinami a ovládacím prvkem pro zobrazení menu s aplikacemi. Aktuálně požadovanými aplikacemi jsou již zmíněné aplikace jako například aplikace MP3 přehrávače, Osvětlení a Informační panel.

1.3 Motivace

Operační systém Android a programování aplikací pro tento operační systém v jazyce Java je vhodným prostředkem k řešení daného problému. Použití tabletu s tímto operačním systémem pro školní vývoj znamená, na rozdíl od existujících systémů v automobilovém průmyslu při případném poškození, efektivnější a levnější opravu v podobě výměny tabletu za nový, do kterého by se pouze nahrály příslušné aplikace.

2 Současný stav

Nynější trh poskytuje širokou nabídku elektromobilů a mezi jejich výbavu patří displej na místo typických ukazatelů. Ne zřídka se jedná dokonce o dotykové displeje. Často bývá použit jeden centrální displej, umístěný uprostřed palubní desky a druhý informační displej za volantem v místě již zmíněných ukazatelů. Centrální displej kromě poskytování informací o stavu vozidla a dalších multimediálních funkcí často řidiči umožňuje další ovládání vozidla. Zatímco informační displej by měl řidiče pouze informovat o typickém základním stavu vozidla, tím je myšlena například rychlost, zařazený rychlostní stupeň, stav baterie, teplota baterie apod.

Je snahou, aby ve školním elektromobilu byly poskytovány informace pomocí tabletu, jakožto levné alternativy pro školní potřeby. Ke znázornění získané inspirace je dále popsáno několik dnešních eleltromobilů. Především jejich displejů a informace, které prostřednictvím displeje poskytují.

2.1 BMW i3

Jedná se o elektrický automobil, který se poprvé objevil jako koncept už v roce 2011 a na trh přišel o dva roky později. Dosud, však zejména u nás, jde o stále raritní a netypické vozidlo. Navíc se jedná o první sériově vyráběný elektromobil od BMW.

Palubní deska je po designové stránce propracovaná, zároveň působí jednoduše. V jejím středu je standartně umístěn centrální displej multimediálního systému s úhlopříčkou 10,2". Jeho vnitřní prostředí tvoří infotainment iDrive, který se ovládá prostřednictvím otočného ovladače umístěného mezi sedadly řidiče a spolujezdce. Práce s ním, i samotné ovládání je díky přítomnosti jeho dotykové plochy rychlá, jednoduchá a intuitivní. Samozřejmě nechybí většina BMW technologií a funkcí, mezi něž patří třeba navigace se zdařilou prostorovou 3D grafikou, či příplatkové připojení k internetu. Speciální pozornost si zaslouží především funkce ConnectedDrive + Services, která prostřednictvím integrované SIM karty, umožňuje mimo jiné kontaktovat infocentrum BMW a bez jakékoliv námahy si po krátkém live hovoru vyhledá nejbližší nabíjecí stanici. Následně nabídne poslání nového cíle do navigace, respektive trasu.

Další displej (obr. 2.1), nahrazující klasické analogové ukazatele, najdeme za dvouramenným volantem. Obrazovka pomocí diagramu nás informuje o množství využívání elektromotoru, respektive rekuperace, v závislosti na síle stlačení plynového pedálu. Nechybí ani ukazatel aktuální rychlosti, ale i aktuálního stavu akumulátorů a informace o dojezdu. Zavolantem se též nachází volič směru jízdy (v pořadí D, P, N, R) spolu se startovacím tlačítkem [1].

Obrázek 2.1: BMW i3 Displej za volantem [2]

2.2 Nissan Leaf

Další elektromobil, kterým byla bakalářská práce nejvíce inspirována je Nissan Leaf. Tento vůz využívá dvojitý digitální ukazatel s menším horním displejem a větším pod ním (obr. 2.2).

Menší displej v levé části zobrazuje ikonky v závislosti na stavu vozidla, hned vedle vás informuje o úrovni ECO, již při jízdě dosahujete s ohledem na styl jízdy. Uprostřed toho displeje se nachází tachometr a po pravé straně je znázorněn aktuální čas a venkovní teplota. Do malého displeje byly po stranách umístěny i ikonky odbočujících ukazatelů.

Obrázek 2.2: Nissan Leaf - Displeje [3]

Větší displej v levé horní části též zobrazuje ikonky znázorňující stav vozidla. V horní části má umístěn ukazatel výkonu, informuje nás o výkonu motoru v závislosti na sešlápnutí plynového pedálu. Pomocí stupnicového zobrazení POWER je zde i stupnice regenerace, ukazující množství elektřiny, která byla regenerována regenerativním brzděním při brzdění motorem. Levá strana nabízí zobrazení teploty baterie a podobné

zobrazení má ukazatel zbývající baterie na pravé straně. Ten však obsahuje i dojezdovou vzdálenost, která je počítána na základě aktuálního stavu nabití baterie a posledního stavu spotřeby energie. Uprostřed tohoto displeje najdeme informační displej, který nám zachycuje různé informace o vozu, například počítadlo uvádějící součet celkově najetých kilometrů, počítač ujeté zdálenosti, zvolený rychlostní stupeň a provozní kontrolky. Nad ním se nachází kontrolka READY, která svítí při připraveném vozu k jízdě.

Na informačním displeji lze také zobrazit odhadovaný čas nabíjení, úsporu energie, průměrnou rychlost, dobu jízdy a upozornění na nadcházející události údržby. K jeho ovládání slouží čtyři tlačítka umístěná na palubní desce vlevo od volantu [4].

2.3 Tesla Model 3

Nejznámější firma Tesla Motors má zásluhu na zhotovení nejnovějšího elektromobilu Tesla Model 3. Jedná se o chystaný první dostupný elektromobil této značky, který má ambice stát se prvním masově rozšířeným elektromobilem [5].

Na palubní desce elektromobilu by se měl přímo uprostřed nacházet pouze 15" displej (obr. 2.3) s vlastním softwarem Tesly na bázi Linuxu. Software je stále aktualizován a je chystaný update s možností autopilota, na němž se snaží Tesla pracovat.

Obrázek 2.3: Tesla Model 3 interiér [6]

3 Cíl práce

K dosažení cíle je potřebné seznámit se s platformou OS Android a programováním pro tento systém v jazyce Java. Dále se seznámit s možností přizpůsobení zařízení s OS Android kioskovému režimu. Je třeba se obeznámit s možnostmi integrace tabletu jako multifunkčního panelu do elektromobilu eŠus vyvíjeného na TUL, případně navrhnout způsoby ovládání funkcí elektromobilu (audio systém, osvětlení, přístup k internetu, čtení informací o palubní síti a další). Důležité je naprogramovat aplikaci pro multifunkční panel, která bude realizovat ovládání alespoň dvou funkcí elektromobilu.

Jako multifunkční panel pro elektromobil bude použit tablet a ten se postupně přizpůsobí potřebám elektromobilu. Jako výchozí aplikaci poskytne ***Zavaděč*** kioskový režim, ze kterého budou přístupné aplikace vytvořené pouze pro potřeby elektromobilu a pro další jeho využití. Samotný operační systém Android zůstane zachován, a je dokonce možné do běžné výchozí plochy OS Android se přepnout.

Mezi základní funkce elektromobilu v kioskovém režimu, které budou vytvořeny, byly zvoleny aplikace ***MP3 přehrávače***, ***Osvětlení*** a ***Informační panel***.

Aplikace ***MP3 přehrávače*** nám umožní přehrávání mp3 souborů z tabletu do v budoucnu připojených vestavěných reproduktorů v elektromobilu pomocí konektoru 3,5 Jack.

Osvětlení je připravovanou aplikací pro budoucí zobrazování rozsvícených a zhasnutých světel elektromobilu. Ta jsou prozatím simulována pomocí obsažených ovládacích prvků pro vysílání signálu ohlašujícího stav jednotlivých světel a jejich zpětnou vazbu. Simulace je prováděna pomocí externího hardwaru.

Tím je UMFT311EV. Jedná se o modul pro rychlý přechod k technologii USB, který umožňuje navrhování koncových produktů. Jeho zapojením je možné snadno získat budoucí informace z elektromobilu a přenést je do tabletu, kde budou pomocí uživatelského rozhraní graficky zobrazena.

Další aplikace ***Informační panel*** poskytne informace, týkající se elektromobilu, pomocí sběrnice CAN. Tato sběrnice je dnes již přítomna v každém automobilu a skze ni putují různé informace o stavu jednotlivých částí vozu. Aplikace by měla být schopna zobrazit především aktuální rychlost, zařazený rychlostní stupeň a stav baterie.

Také proběhne pokus o root tabletu, který by odblokoval rozšiřující možnosti v operačním systému Android. Tato operace by podpořila budoucí vývoj aplikací pro elektromobil.

4 Návrh řešení

Nejprve bude potřeba seznámit se s programováním pro platformu Android a zjistit další informace pro použití tabletu jako kioskového režimu v systému Android.

4.1 Tablet

Pro účel bakalářské práce bude použit tablet ***ASUS MeMO Pad 8***. Díky svému osmipalcovému Full HD displeji s rozlišením 1920 × 1200 pixelů a širokým pozorovacím úhlem 174 stupňů poskytne uživateli skvělý obraz. Také jeho čtyřjádrový 64-bitový procesor Intel Atom Quad-Core Z3560 a 2GB RAM zajistí plynulé ovládání tabletu i během spuštění více aplikací najednou. Výdrž baterie tabletu se pohybuje kolem dostatečných 9 hodin. Obsahuje operační systém Android 4.4 a mimo jiné poskytne i Bluetooth 4.0, GPS a vysokorychlostní internet LTE.

4.2 Nastavení pro kioskový režim

Pro zjednodušení je nutné omezit dostupnost některých funkcí, jež poskytuje OS Android. Aplikace určené pro elektromobil je dobré zobrazit přes celou plochu displeje, tím bude skryta horní ovládací lišta a nebude narušovat pozornost řidiče. Dále je třeba zamezit opuštění spuštěné aplikace s kioskovým režimem, tedy přepnutí do výchozí plochy běžného OS Android.

Tablet je potřeba i patřičně nastavit. Jelikož bude zabudován v elektromobilu, nebude možné jej probrat z režimu spánku bočním tlačítkem. Tento problém lze vyřešit vypnutím spořící obrazovky a ponecháním stále zapnutého displeje. Stačí ve výchozí ploše běžného OS Android otevřít ***Nastavení***, v sekci zařízení zvolit ***Displej***, následně zvolit ***Režim spánku***, kde se nastaví volba ***Bez přestávky***. Též zde zvolit ***Spořič obrazovky*** a vypnout jej pomocí horního posuvníku.

Pokud bude využito v běžném OS Android dalších aplikací, je dobré vědět, že například v případě stálého přihlášení k aplikaci facebook s připojením k internetu, mohou později příchozí zprávy svým zvukem vyrušovat v kioskovém režimu, neboť aplikace v OS Android stále běží a jsou pouze v pozadí za již zmíněnou aplikací s kioskovým režimem. Má-li být tomuto jevu zabráněno, postačí příslušným účtům vypnout synchronizaci. To lze provést v běžném OS Android v ***Nastavení***, konkrétně v sekci účty a po zvolení daného účtu vypnout synchronizaci.

4.3 Zavaděč

Aplikace spuštěná hned po startu tabletu, tvořící kioskový režim, nese jméno Zavaděč. Tato aplikace se bude skládat ze dvou pohledů, jedním bude úvodní obrazovka a druhý poskytne seznam dalších aplikací v tabletu.

Spuštění hned po startu bude provedeno, pokud je v ***Nastavení*** v sekci zařízení volbou ***Plocha*** zvolena aplikace Zavaděč. Taktéž, pokud bude třeba se přepnout do výchozí plochy běžného OS Android, lze zde zvolit původní aplikaci. Tuto volbu je však před vypnutím tabletu nutné vrátit, pokud má být při spuštění tabletu zobrazen Zavaděč.

Úvodní obrazovka zobrazí logo elektromobilu, název elektromobilu, aktuální čas a ovládací prvek pro zobrazení seznamu aplikací. Tento prvek bude aktivní skoro přes celou polovinu displeje, tedy bude reagovat po této oblasti, samotný obrázek ovládacího prvku bude zakrývat pouze menší část ve středu této oblasti. Aplikace obsažené v seznamu aplikací mají být pouze aplikacemi potřebnými k práci s elektromobilem. To bude zajištěno filtrací na základě textového řetězce v názvu balíčku jednotlivých aplikací.

Zavaděč bude potřeba odlišit od klasických aplikací, neboť bude využit jako úvodní aplikace. Také je potřebné zajistit, aby byla spuštěna pouze jedna instance aplikace jím a z něj spuštěná. Dále bude zapotřebí vykreslit dva pohledy, pro úvodní displej aplikace a pro seznam aplikací.

4.4 MP3 přehrávač

Přehrávání hudebních souborů bude obsahovat též dva pohledy, jeden s ovládacími prvky a druhý se seznamem skladeb.

Přehrávač bude obsahovat název přehrávané skladby, ovládací prvky, posuvník pro posuv aktuálního času přehrávání a posuvník k ovládání hlasitosti. Mezi ovládací prvky budou patřit funkce ***spustit***, ***pozastavit***, ***další***, ***předchozí***, ***posun vpřed***, ***posun vzad***, ***opakování skladby***, ***náhodné přehrávání*** a ***seznam skladeb***.

Název skladby kopíruje název souboru bez koncového řetězce typu „.mp3“. Funkce ***spustit*** spustí přehrávání aktuální skladby. Po spuštění skladby se změní ve funkci ***pozastavit***, jež pozastaví skladbu na dané pozici. Funkce ***další*** poslouží k přepnutí se na další skladbu, obdobně funkce ***předchozí*** spustí předcházející skladbu. Ke krátkému posunu ve skladbě bude možno použít funkce ***posun vzad*** a ***posun vpřed***, které posunou skladbu o 5 vteřin daným směrem. Funkce ***opakovat skladbu***, přehrává jednu skladbu stále dokola na rozdíl od funkce ***náhodné přehrávání*** určené k náhodnému pořadí přehrávaných skladeb. A funkce ***seznam skladeb***, zobrazí seznam skladeb. Nejedná se o nic jiného než o seznam skladeb obsažených v tabletu. Seznam skladeb je tvořen hlavičkou s textem „Seznam skladeb“ a samotným aktivním seznamem skladeb. Kliknutím na požadovanou skladbu se skladba začne přehrávat.

4.5 Osvětlení

Tato aplikace v budoucnu poslouží pouze jako grafické znázornění rozsvícených a zhasnutých světel elektromobilu, nyní však bude obsahovat i ovládací prvky ke snadné simulaci skrze externí hardware UMFT311EV.

Pomocí ovládacích prvků budou ovládána přední a zadní světla, dálková světla, pravý a levý blinkr, výstražný trojúhelník a brzdy. Tato světla budou specificky graficky znázorněna. Například v pravém horním rohu bude graficky znázorněn stav předního pravého světla (i dálkového pravého světla) a pravý blinkr. Obdobně v levém dolním rohu stav levého světla, levého blinkru a levé světlo brzd.

4.6 Informační panel

Díky této aplikaci by nám mělo být umožněno grafické zobrazení informačních hodnot získaných z elektromobilu pomocí sběrnice CAN. Zatím však neexistuje žádné fyzické propojení tabletu se sběrnicí CAN ve školním elektromobilu. Proto bude vytvořeno především grafické znázornění některých informací, které bude v budoucnu možné z elektromobilu vyčíst.

Námi připravovaná aplikace by měla mít dvě různé plochy. Počáteční plocha obsahující všeobecné základní informace o elekromobilu v podobě například zařazeného rychlostního stupně, aktuální rychlosti, stavu nabití baterie, popřípadě zobrazen odhadovaný dojezd na baterii a teplotu baterií. Též by zde neměly chybět najeté kilometry a možnost měření trasy. Druhá plocha by nám pak umožnila zobrazení bližších informací o baterii např., zda je připojena k dobíjecímu zařízení, přesné procento vybití a stupeň teploty baterie.

Strukturu aplikace chceme vytvořit tak, aby bylo možné aplikaci dále rozšiřovat a vyvíjet. Tedy připravit se na možnost různého typu zdroje pro snadnou aktualizaci aplikace. Je třeba mít též možnost přijatá data zpracovávat pomocí rozdílných operací a ty zpracované dále posílat do grafického rozhraní. Lze tedy říci, že se bude jednat o tři vrstvy, které budou tvořit zdroje, operace a grafika.

4.7 Odblokování systému

Pokud se podaří provést odblokování systému zařízení, zpřístupní se další možnosti využitelnosti zařízení. Avšak běžnému uživateli jsou výrobcem práva omezena, neboť s právy superuživatele je snadné zařízení znehodnotit. Proto také po operaci odblokování systému zařízení, obvykle ztrácíme záruku. Pro bližší představu, superuživatel je v podstatě přirovnatelný k administrátorovi v OS Windows. Takovýto administrátor má práva instalovat libovolné aplikace a mazat libovolná data.

Bude hledán případný postup řešení, kterým lze tablet odblokovat. Podle tohoto návodu by dále proběhl postup k cílenému odblokování tabletu a získání tak širších možností pro budoucí vývoj spojený s rozvojem školního elektromobilu.

5 Realizace řešení

Pro programování na platformě Android, bylo nezbytné nainstalovat vývojové prostředí. Využilo se nově vzniklého vývojové prostředí Android Studio. Pro snadnou dostupnost a větší efektivitu byly problémy při programování řešeny za využití internetového zdroje, konkrétně webu Android Developers určeného přímo pro Android vývojáře. Odkazy na příslušné kapitoly internetové dokumentace, ze kterých bylo čerpáno, jsou uvedeny v textu.

Dle vyhledaných informací byl nastaven systém Android pro potřeby kioskového režimu a všechny aplikace vytvořeny s výchozími parametry zmíněnými níže.

Obrázek 5.1: tablet v eŠusu

Aby bylo zabráněno rozptylování řidiče elektromobilu notifikační lištou OS Adnroid, jsou vytvořené aplikace zobrazeny přes celou plochu displeje (obr. 5.1). K tomu přispěl konkrétní řádek kódu, popřípadě dva řádky kódu vložené do každé aktivity v souboru ***AndroidManifest.xml*** (kód 5.1).

Ukázka kódu 5.1: Skrytí notifikační lišty

```
<activity
android:theme="@android:style/Theme.Wallpaper.NoTitleBar.Fullscreen"
android:configChanges="orientation|keyboardHidden|screenSize" >
```

5.1 Zavaděč

Aplikace zavaděče je odlišná především ve způsobu spuštění. Nejedná se totiž a typickou spustitelnou aplikaci z kategorie ***Launcher***, ale je zařazena v kategoriích ***Home*** a ***Default***.

5.1.1 AndroidManifest

Aby zavaděč fungoval dle představ, bylo potřeba do souboru AndroidManifest.xml přidat práva na ohlášení dokončeného boot zařízení, čtení a zápisu na externí uložiště (kód 5.2) [7].

Ukázka kódu 5.2: Práva týkající se boot zařízení, čtení a zápisu na externí uložiště

```
<uses-permission
   android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission
   android:name="android.permission.WRITE_EXTERNAL_STORAGE "/>
<uses-permission
   android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
```

První dvě dovolují čtení a zápis do externího uložiště. Třetí ohlásí boot tabletu, a hned potom umožní spuštění Zavaděče. Dále byla do Zavaděče, pro vytvoření pouze jedné instance spuštěného procesu, přidána primární aktivitě tato vlastnost (kód 5.3) [8].

Ukázka kódu 5.3: Vytvoření jedné instance

```
<aktivity android:launchMode="singleTask" >
```

Aby si aplikace spuštěné ze Zavaděče uchovávaly jen jednu činnost, ke které se při následném vrácení do aplikace vrátí, byla použita podobná vlastnost pro aktivitu seznamu aplikací (kód 5.4).

Ukázka kódu 5.4: Urdžení jedné instance

```
<aktivity android:launchMode="singleInstance" >
```

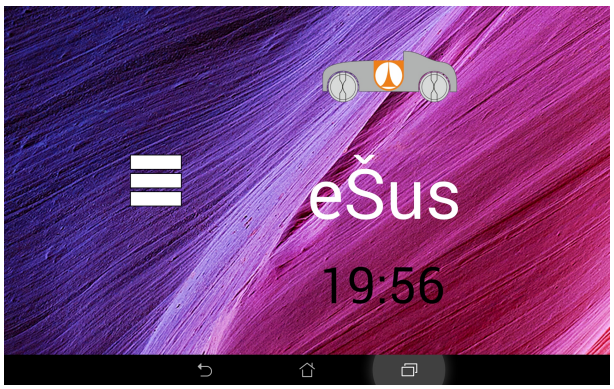
A v poslední řadě byly přiřazeny již výše zmíněné kategorie ***Home*** a ***Default*** k hlavní aktivitě vložením do této aktivity (kód 5.5) [9].

Ukázka kódu 5.5: Přiřazení kategorií

```
<intent-filter>
   <action android:name="android.intent.action.MAIN" />
   <category android:name="android.intent.category.HOME" />
   <category android:name="android.intent.category.DEFAULT" />
</intent-filter>
```

5.1.2 HomeActivity

Třída ***HomeActivity*** zahrnuje metodu ***onCreate***, kde je nastaveno přiřazení k souboru ***activity_home.xml*** a ten se stará o grafické rozhraní hlavní aktivity (obr. 5.2).



Obrázek 5.2: Zavaděč

Dále v této třídě existuje jednoduchá funkce, jež spustí a zobrazí jinou aktivitu obsahující seznam aplikací (kód 5.6).

Ukázka kódu 5.6: Zobrazení aplikací

```
public void showApps(View v){
   Intent i = new Intent(this, AppsListActivity.class); startActivity(i);
}
```

Protože je nežádoucí, aby docházelo k možnému opuštění aplikace zavaděče do běžného režimu OS Android, ošetříme tlačítko ***Zpět***. A to tak, že je mu nastavena stejná funkce jako tlačítku ***Domů*** (kód 5.7) [10].

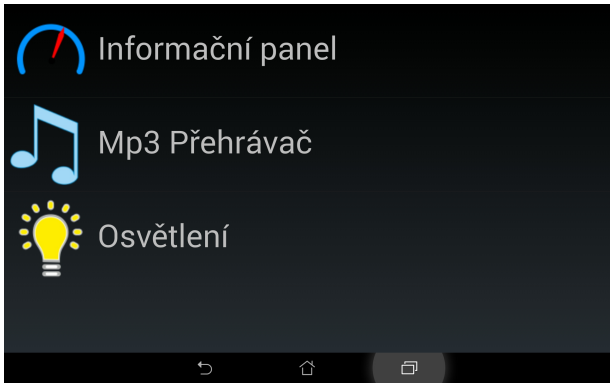
Ukázka kódu 5.7: Zabránění opuštění aplikace

```
@Override
public void onBackPressed() {
   Log.d("CDA", "onBackPressed Called");
   Intent setIntent = new Intent(Intent.ACTION_MAIN);
   setIntent.addCategory(Intent.CATEGORY_HOME);
   setIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
      startActivity(setIntent);
}
```

5.1.3 AppListActivity

Ve třídě, týkající se aktivity seznamu aplikací, jsou v metodě ***onCreate*** přiřazení k souboru ***aktivity_apps_list.xml*** s grafickým rozhraním (obr. 5.3) a voláné tři funkce.

První funkce nazvaná ***loadApps*** pomocí balíčkového manažeru načítá aplikace v tabletu, které před přidáním do seznamu aplikací filtruje na aplikace, jejichž balíček za-



Obrázek 5.3: Menu v zavaděči

číná textovým řetězcem „cz.tul.“, posléze jsou takto načteny pouze aplikace pro potřeby elektromobilu. Ty samozřejmě musely být nejprve naistalovány do tabletu. Dalším budoucím aplikacím tedy postačí, aby název balíčku začínal tímto řetězcem. V případě dodržení tohoto pravidla bude aplikace obsažena v seznamu aplikací vytvořeného Zavaděče.

Druhá funkce nazvaná ***loadListView*** vytváří grafické rozvržení seznamu aplikací. Využívá jednoduchého grafického rozmístění jednotlivých částí položky seznamu ze souboru ***list_item.xml***.

Poslední funkcí je ***addClickListener*** pro naslouchání a následné spuštění aplikace ze seznamu po jejím zvolení. Protože každá položka seznamu obsahuje více než jednu informaci, bylo vhodné vytvořit třídu nazvanou ***AppDetail***. Ta je tvořena názvem balíčku, názvem aplikace a ikonou aplikace.

5.2 MP3 přehrávač

Pro čtení souborů mp3 z tabletu bylo zapotřebí do souboru ***AndroidManifest.xml*** přidat práva ke čtení z externího uložiště.

5.2.1 MainActivity

Třída MainActivity implementuje metody nazvané ***OnCompletionListener*** a ***SeekBar.OnSeekBarChangeListener.OnCompetionListener***, která hlídá a informuje o dokončení přehrávání mp3 souboru. ***SeekBar.OnSeekBarChangeListener*** je metoda upozorňující na změnu posuvníku.

Začátek třídy obsahuje deklaraci veškerých ovládacích prvků a proměnných, které jsou následně použity. V metodě ***onCreate*** je nastaveno připojení k souboru ***player.xml*** s grafickým rozhraním (obr. 5.4), dále jsou zde nastaveny veškeré ovládací prvky přehrávače a několik dalších vlastností jako například překreslení po otočení displeje, několik různých naslouchačů a další.

Funkce nazvaná ***onActivityResult***, vrací index ze seznamu skladeb a přehrávanou skladbu. Samotné spuštění přehrávání skladby zajišťuje funkce ***playSong***, v níž jsou ta-

Obrázek 5.4: MP3 přehrávač

ké nastaveny stavy příslušných prvků. Další funkce ***updateProgressBar*** se zabývá aktualizací posuvníku indikujícího čas přehrávání. V rámci procesu byla vytvořena nezávislá funkce, která udržuje a nastavuje celkový čas přehrávané skladby a již uběhlý čas skladby. Poté následuje funkce nazvaná ***onStartTrackingTouch*** reagující na uživatelské spuštění přehrávání skladby a spustí též posuvník indikující přehrávání skladby. Naopak funkce nazvaná ***onStopTrackingTouch*** reaguje na pozastavení vyvolané uživatelem a tentýž posuvník pozastaví na aktuální pozici.

K opakování skladby, či náhodnému přehrávání skladeb je určena funkce s názvem ***onCompletion***, která po přehrání skladby na základě uživatelského nastavení řeší spuštění další skladby. Tedy zda spustit tutéž skladbu znovu, přehrát náhodnou další nebo obyčejně přehrát následující skladbu v seznamu. Uvolnění přehrávané skladby pak vyřeší funkce nazvaná ***onDestroy***.

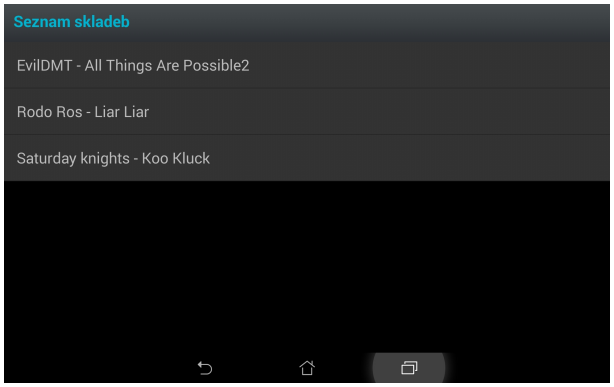
Jelikož při otočení obrazovky nedocházelo k ideálnímu grafickému rozložení, byla použita funkce s názvem ***onConfigurationChanged***, jenž zajišťuje správné zobrazení v dané pozici otočeného tabletu. Protože opuštěním aplikace MP3 přehrávače docházelo k ukončení procesu, tudíž k ukončení přehrávání skladby, bylo použito nahrazení tlačítka ***Zpět*** tlačítkem ***Domů***. Obdobný problém je již vyřešen u aplikace zavaděče.

5.2.2 PlayListActivity

Tato třída PlayListActivity nedědí od standartní třídy Activity, ale od standartní třídy ListActivity. Uvnitř třídy je hned zpočátku vytvořena veřejná proměnná typu ArrayList, která bude obsahovat seznam mp3 souborů, jejich názvy a cesty k souborům. Do ní jsou posléze načteny jednotlivé mp3 soubory. Metoda ***onCreate*** obsahuje nastavení připojení k souboru ***playlist.xml*** s grafickým rozhraním (obr. 5.5). Navíc je v ní mimo jiné řešeno načtení mp3 souborů z externího uložiště, přidání mp3 souborů jako jednotlivých položek menu do ***ListView*** a hlídání výběru jednotlivých položek.

5.2.3 SongManager

SongManager je třída, jež je využita k vyhledávání mp3 souborů na externím uložišti. Má vytvořenu finální proměnnou s řetězcem představujícím cestu k externímu uložišti.



Obrázek 5.5: playlist MP3 přehrávače

Za ní následuje proměnná songsList totožná s předchozí u třídy ***PlayListActivity***. Dále je zde funkce nazvaná ***searchMapsPath***, v níž je kontrolována existence adresáře a následně jsou vyhledávány mp3 soubory.

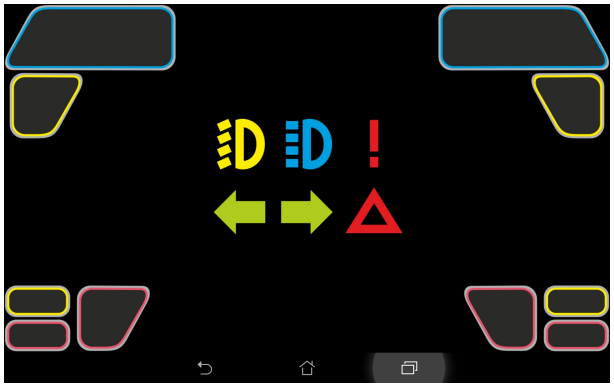
Funkce s názvem ***getPlaylist*** nám pak vrací vytvořený seznam skladeb, ve kterém je název a cesta k příslušnému mp3 souboru. Obsažena je zde dokonce i třída filtrující soubory k získání pouze souborů s příponou „.mp3“ či „.MP3“.

5.2.4 Utilities

Do třídy ***Utilities*** byli vepsány funkce nazvané ***milliSecondsToTimer***, ***getProgressPercentage*** a ***progressToTimer***. Funkce ***milliSecondsToTimer*** nám převádí milisekundy do časového formátu hodiny:minuty:sekundy. Návratová hodnota následující funkce ***getProgressPercentage*** představuje aktuální hodnotu posuvníku v procentech. A poslední funkce ***progressToTimer*** vrací aktuální hodnotu posuvníku převedenou na milisekundy.

5.3 Osvětlení

Díky již vytvořeným názorným demo aplikacím pro externí harware UMFT311EV, bude využito aplikace GPIODemo, která bude upravena. Ta mimo jiné obsahuje soubor ***FT311GPIOInterface.java***, s jehož pomocí budou ovládány vstupy a výstupy obvodu FTDI. Dále byl vytvořen pohled, tedy názorné grafické rozhraní pro znázornění rozsvícených a zhasnutých světel elektromobilu s ovládacími prvky, určenými pouze k simulaci (obr. 5.6). Také byl vytvořen soubor se stejnojmennou třídou aktivity, za účelem zobrazení stavu a ovládání světel, nazvané ***LightsControlActivity***, dědící od standartní třídy ***Activity***. V ní je předdefinovaná metoda ***onCreate*** pro inicializaci aktivity. Třída též obsahuje funkce pro nastavení dat ***ConfigData***, načtení dat ***ReadData*** a vynulování dat UMFT311EV ***resetFT311***. V poslední řadě obsahuje také funkci k vykonání procesu čtení dat ***ProcesReadData***.



Obrázek 5.6: Osvětlení

5.3.1 UMFT311EV

UMFT311EV (obr. 5.7) je vývojový modul, který využívá integrovaný obvod FTDI k rozvoji USB příslušenství připojených k zařízením s platformou Android. Jedná se o USB hostitele s plnou rychlostí speciálně zaměřeného na poskytování přístupu k periferním hardwaru z platformy Android přes USB port zařízení. Tento modul přenáší data do uživatelského rozhraní.

Obrázek 5.7: UMFT311EV

5.3.2 AndroidManifest

Do souboru AndroidManifest.xml byly kvůli použití externího hardwaru přidány následující řádky (kód 5.8) [11].

Ukázka kódu 5.8: USB s externím hardwarem

```
<uses-feature android:name="android.hardware.usb.accessory" />
```

Dále byly do jediné aktivity v tomto souboru přidány ještě řádky (kód 5.9).

Ukázka kódu 5.9: Připojení UMFT311EV

```
<intent-filter>
   <action
      android:name="android.hardware.usb.action.USB_ACCESSORY_ATTACHED"
   />
</intent-filter>

<meta-data
   android:name="android.hardware.usb.action.USB_ACCESSORY_ATTACHED"
   android:resource="@xml/accessory_filter">
</meta-data>
```

5.3.3 LightsControlActivity

Třída ***LightsControlActivity*** zahrnuje deklaraci všech potřebných proměnných a následně v metodě ***onCreate*** těmto proměnným nastaví hodnoty a další vlastnosti. Dále obsahuje funkci ***ConfigData*** pro nastavení příslušných hodnot do proměnných a také na příslušný port. Funkce ***ReadData*** se stará o načtení hodnot z příslušného portu a posílá je k dalšímu zpracování. To je zajištěno funkcí ***ProgressReadData***, v níž už dochází nejen ke změnám proměnných, ale i k projevu grafického znázornění stavu jednotlivých vstupů. Za zmínku stojí i funkce s názvem ***resetFT311***, která nastaví defaultní hodnoty portu a zároveň ovládacím prvkům a proměnným.

5.4 Informační panel

Zde byly v původním balíčku vytvořeny další balíčky a to ***components***, ***core*** a ***datasource***. Struktura balíčků odráží strukturu vzájemně nezávislých vrstev propojených přes jednoduchá rozhraní. Balíček ***components*** obsahuje námi vytvořené komponenty. V balíčku ***core*** se nachází struktura jádra, které provádí operace se zdrojovými vstupy a posílá je na grafické výstupy. A balíček ***datasource*** má v sobě třídy pro různé zdroje. Původní balíček obsahuje také dvě aktivity ***DefaultDisplayActivity*** a ***BatteryStatusDisplayActivity***.

5.4.1 AndroidManifest

Pro simulaci odpojení či zapojení dobíjecího zařízení bylo využito samotného tabletu a jeho dobíjení. Ke zjištění stavu připojení USB konetoru jsme museli do souboru ***AndroidManifest.xml*** vložit příslušné řádky (kód 5.10) [12].

Ukázka kódu 5.10: Stav připojení USB

```
<receiver android:name=".datasource.TabletPowerConnectionReceiver">
   <intent-filter>
      <action
         android:name="android.intent.action.ACTION_POWER_CONNECTED"/>
```

```
        <action
            android:name="android.intent.action.ACTION_POWER_DISCONNECTED"/>
    </intent-filter>
</receiver>
```

5.4.2 DefaultDisplayActivity

DefaultDisplayActivity aktuálně obsahuje dvě statické proměnné, které představují zdroje. Jedna je typu ***Simulation*** a vyvolává simulační hodnoty pro simulaci aktivity jednotlivých grafických prvků. Druhou představuje typ ***CAN***, který je už schopen ze získaného rámce na sběrnici CAN vyčíst jednotlivé informace. Tyto rámce jsou zatím pouze simulovány na vstupu, neboť prozatím neexistuje fyzické propojení tabletu se školním elektromobilem. V metodě ***onCreate*** je vytvořeno propojení s komponentami pro zobrazení stavu baterie a zařazeného rychlostního stupně, aby na ně mohl být aplikován ***setOnClickListener***. Ten provede přiřazenou fukci po kliknutí na komponentu. Následně byla v metodě ***onPostCreate*** vytvořena inicializace zdrojů.

5.4.3 BatterySatusDisplayActivity

V této aktivitě byly v metodě ***onCreate*** propojeny a inicializovány proměnné s informacemi o připojeném napájení, procentuálním stavu baterie a teplotním stupni baterie. Dále totéž v komponentě pro zobrazení stavu baterie. V aktivitě byl vytvořen ***ValueChangeListener*** (kód 5.11), který hlídá aktuální stav proměnných dle zdrojových hodnot.

Ukázka kódu 5.11: Hlídání stavu proměnných

```
ValueChangeListener listener = new ValueChangeListener() {
  @Override
  public void valueChanged(final String name, final String value) {
     BatteryStatusDisplayActivity.this.bPercentilValue.post(new
        Runnable() {
        @Override
         public void run() {
            if (name.equals("tabletBatteryIsCharging")) {
               isCharging = value;
            }
            if (name.equals("tabletBatteryLevel")) {
               batteryLevel = value;
            }
            if (isCharging.equals("true")) {
               BatteryStatusDisplayActivity.this.bChargeValue
.setText("ZAPOJENO");
            } else {
               BatteryStatusDisplayActivity.this.bChargeValue
.setText("ODPOJENO");
            }
```

```
                BatteryStatusDisplayActivity.this.bPercentilValue
.setText(batteryLevel + " %");
                BatteryStatusDisplayActivity.this.bTemperatureValue
.setText(baterryTemperature + " °C");
            }
        });
    }
};
```

A v metodě ***onPostCreate*** je inicializované jádro (kód 5.12), které zjišťuje a nastavuje aktuální hodnoty zdrojů do proměnných.

Ukázka kódu 5.12: Inicializace jádra

```
Core c = CoreFactory.getCore("Simple");
if (c != null){
    isCharging = c.getValue("tabletBatteryIsCharging");
    batteryLevel = c.getValue("tabletBatteryLevel");
    if (isCharging.equals("true")) {
        bChargeValue.setText("ZAPOJENO");
    } else {
        bChargeValue.setText("ODPOJENO");
    }
    c.addValueChangeListener(listener);
}
```

5.4.4 Komponenty

Komponenty vykreslují hodnody poskytnuté jádrem do UI. Mezi vytvořené komponenty patří ***analogový indikátor*** (pro zobrazení aktuální rychlosti), ***Indikátor baterie*** (zobrazující stav baterie a odhadovaný dojezd), ***digitální indikátor*** (zobrazuje zvolený rychlostní stupeň, najeté kilometry a měření naší cesty) a ***indikátor teploty baterie*** (zobrazující teplotu baterie). Všechny komponenty jsou využity hned na defaultní ploše aplikace. Na ploše stavu baterie jsou pak pouze komponenta indikace stavu baterie a další informace zda je připojeno napájení, přesné procento nabití baterie a teplota baterie. Aplikace byla vytvořena s responzivním designem (obr. 5.8).

5.4.5 Jádro

Jádro slouží ke zpracování dat ze zdrojů a zpracové hodnoty poskytuje komponentám na UI. V aplikaci je implementováno jednoduché jádro označené ***SimpleCore***, které pouze ukládá hodnoty poskytnuté zdroji. Struktura jádra je tvořena tak, aby její rozšiřování a další vývoj měl minimální následky na změnu zdrojového kódu. Je ideální, aby byl zdrojový kód opravdu pouze rozšířen. Bylo vhodné použít interface pro jádro (kód 5.13). Ten obsahuje metody pro nastavení hodnoty, získání hodnoty, vyčištění hodnot, přidání ***ValueChangeListeneru***, jeho odebrání a vyčištění. Znázornění propojení jednotlivých tříd, pomocí UML Class Diagramu naleznete v příloze A.

Obrázek 5.8: Responzivní design aplikace

Ukázka kódu 5.13: Interface jádra

```
public interface Core {
    void setValue(String name, String value);
    String getValue(String name);
    void clearValues();
    void addValueChangeListener(ValueChangeListener listener);
    void removeValueChangeListener(ValueChangeListener listener);
    void clearValueChangeListener();
}
```

Dále byla vytvořena třída ***CoreFatory*** pro jádra (kód 5.14) obsahující hash tabulku jader a zajišťující jejich načtení. Třída poskytuje práci s konkrétním jádrem napříč celou aplikací. Jádro je identifikováno svým názvem.

Ukázka kódu 5.14: Zvolení jádra

```
public class CoreFactory {
    private static Hashtable<String, Core> cores = new Hashtable<>();
    public static Core getCore(String coreName) {
        switch (coreName) {
            case "Simple": {
                if (!cores.containsKey(coreName)){
                    cores.put("Simple", new SimpleCore());
                }
```

```
            return cores.get("Simple");
        }
        default: return null;
    }
  }
}
```

Následně bylo vytvořeno samotné jádro a v budoucnu lze vytvořit další s pokročilejším zpracováním poskytovaných dat zdroji (např. dojezd elektromobilu z aktuálních hodnot a statistik uložených v jádře). K aktuálně potřebným účelům postačilo jedno jádro. Do hash tabulky se v něm zapisují zjištěné hodnoty zdrojů a vytvářejí se k nim posluchače, jež ukládáme do ***ArrayListu listenerů***. V jádře jsou metody pro nastavení hodnoty (kód 5.15), získání hodnoty, vyčištění hodnoty, přidání ***ValueChangeListeneru***, jeho odebrání a vyčištění. Všechny metody jsou synchronizovány.

Ukázka kódu 5.15: Nastavení hodnoty

```
@Override
public synchronized void setValue(String name, String value) {
    String oldValue = table.get(name);
    Boolean changed = (oldValue == null || oldValue != value);
    table.put(name, value);
    if (changed) {
        for (ValueChangeListener l:listeners
                ) {
            l.valueChanged(name, value);
        }
    }
}
```

V balíčku jádra je též interface pro ValueChangeListener, který obsahuje pouze metodu pro změnu hodnoty.

5.4.6 Zdroje

Tento balíček obsahuje třídy jednotlivých zdrojů, které poskytují své hodnoty jádru. Zdroje jsou aktuálně tvořeny ze ***Simulation***, ***CAN*** a ***TabletPowerConnectionReceiver***. Třída ***Simulation*** v časovém úseku mění simulační hodnoty zařazeného rychlostního stupně a aktuální rychlosti vozu. Dále inicializuje třídu ***TabletPowerConnectionReceiver***, která se stará o zjištění stavu připojení USB konektoru a stavu nabití baterie. Obě třídy obsahují metodu ***init***, která inicializuje a nastaví hodnoty do našeho jádra.

Třída ***CAN*** obsahuje třídu ***CANFrame***, která zpracovává získaný rámec s daty. Data jsou simulavána na základě naměřených hodnot zjištěných z Diplomové práce [13]. Cyklicky se mění v nastaveném časovém úseku, neboť zatím neexistuje fyzické propojení tabletu s elerktromobilem.

Tabulka 5.1: Převedené hodnoty rámců (hex., nevypsané byty = 0)

Druh	Stav	byte 0	byte 1	byte 2	byte 3	byte 5	byte 6
Volící páka	P	44	80	30	60	0	0
	R	44	80	2E	40	0	0
	N	44	80	2C	40	0	0
	D	44	80	2A	40	0	0
	S	44	80	38	40	0	0
	+-	44	80	3C	40	0	0
	+	44	80	34	40	0	0
	-	44	80	36	40	0	0
	N s b. ped.	44	80	2C	60	0	0
	P s b. ped.	44	80	30	40	0	0
	inicializace	44	80	21	0	0	0
	inicializace 1	44	80	31	60	0	0
Brzd. ped.	sešlápnutá	72	80	80	0	15	80
	vyšlápnutá	72	80	80	0	05	80
Ruční brzda	poz. horní	32	0	20	40	0	0
	poz. dolní	32	0	20	0	0	0
BMS napětí		03	0	40	0	0	0
BMS SOC		6B	20	40	0	0	0

Tabulka 5.2: Příklad výstupních hodnot

Druh	Stav	ID	pozice bytu	data
Volící páka	P	448	0	83
	R	448	0	72
	N	448	0	62
	D	448	0	52
	S	448	0	C2

Každá hodnota rámce byla převedena do 16 po sobě jdoucích bytů (tab. 5.1), z nichž metoda ***CANFrame*** vychází. Metoda ***CANFrame*** rozděluje rámec na id rámce, pořadí bytu s daty a samotná data (tab. 5.2). Ty poté zpracuje a nastaví do příslušných hodnot jádra.

5.5 Odblokování systému

Bylo nalezeno několik postupů řešení pro ROOT použitého tabletu. Před samotným nálezem postupu se bohužel podařilo odklepnout instalaci novější verze firmwaru, což bylo příčinnou nezdařených pokusů.

První zkoušený postup byl proveden pomocí aplikace ***RootZenFone.apk*** ve verzi ***1.6.6r***, později byla zkoušena i ve verzi ***1.6.8r***. Aplikaci stačí nainstalovat s povolením instalace z neznámých zdrojů. A to v ***Nastavení*** v sekci osobní, zvolením položky ***Zabezpečení***, dále v sekci správa zařízení, zvolením položky ***Neznámé zdroje*** a ***Povolit in-***

stalaci z neznámých zdrojů. Před spuštěním této aplikace bylo potřeba vypnout jakýkoli přístup k internetu, tedy vypnout WiFi a mobilní data. Po spuštění aplikace zvolit jediné tlačítko a dále potvrdit spuštění demo aplikace. Poté ukončit demo aplikaci a restartovat tablet. Znovu spuštěný tablet by následně obsahoval aplikaci superuser, ta by poskytla další nastavení pro superuživatele. Avšak byla zobrazena chyba již po spuštění aplikace ***RootZenFone.apk*** a zvolením tlačítka k provedení operace odblokování systému.

Druhý pokus spočíval ve stažení a instalaci aplikace ***Kingo ROOT*** do počítače a po připojení tabletu se jednoduše pokračovalo v této aplikaci volbou tlačítka další. Bohužel aplikace nepodporovala tablet. Byly hledány další podobné aplikace, ale žádná z nich tablet nepodporovala, a tudíž nebyly ani instalovány.

Později se podařilo snížit verzi softwaru v tabletu pouze o jednu úroveň níže. Byly opakovány tedy zmíněné postupy, bohužel se stejným výsledkem.

Nakonec se na foru XDA developers podařilo najít vhodné řešení problému. Aktuální firmware ve verzi ***.77*** v tabletu byla snížena na verzi ***.57***, v níž operace ROOT nečinila problémy. Se staženými přiloženými soubory byl postup následující:

1. Byla vytvořena záloha tabletu.

2. Tablet byl připojen k počitači, dále zkopírován firmware verze ***.57*** na SD kartu a ten byl přejmenován na ***MOFD_SDUPDATE***.

3. Byl proveden tovární restart, pomocí podržení tlačítka zvýšení hlasitosti spolu s tlačítkem pro spuštění na vypnutém tabletu a spustil se droidboot.

4. Provedl se flash souborů příkazy ***fastboot flash token bom-token.bin***, ***fastboot flash dnx dnx_fwr.bin***, ***fastboot flash dnx dnx_osr.bin***, ***fastboot flash dnx ifwi.bin***, ***fastboot flash radio radio_firmware.bin***, ***fastboot flash tlv update.tlv***.

5. Pomocí tlačítek ovládání hlasitosti byla dále zvolena volba ***sd update*** a potvrzena tlačítkem spuštění. Následně byl z tabletu odpojen USB kabel. Tabletu zčernala obrazovka.

6. V počítači byl spuštěn program ***Intel phone flash tool*** a zvolena ***custom flash***. Vlevo nahoře zvoleno ***systemblankphone*** a kliknuto na předchozí flashnuté soubory. Zahrnuly se do řádků další soubory ***fastboot.img.pos.bin*** a ***boot.img***. Zaškrtla se volba ***on-demand flash mode*** a stiskl ***start flash***.

7. Byla přidržena obě tlačítka pro ovládání hlasitosti a zároveň připojeno USB k tabletu. Dále byl proveden nepředvídatelný čin, ale nedošlo při něm k znehodnocení tabletu. Přibližně po třech minutách byl z tabletu odpojen USB kabel a zastaven flash. Odškrtlo se vše a zůstala pouze zaškrtnutá volba ***only OS Image***. Znovu byl stisknut ***start flash***, po dvou minutách znovu odpojen USB kabel z tabletu a zastaven flash.

8. Následně bylo deset vteřin drženo tlačítko spuštění a připojen USB kabel. Tablet se začal nabíjet a zapnul se displej. Bylo stisknuto tlačítko spuštění tabletu a začalo probíhat snížení verze firmwaru na verzi ***.57***

9. Po té byla nainstalována aplikace ***RootZenFone.apk*** ve verzi ***1.6.8r***, zapnut režim letadla a úspěšně proveden ROOT tabletu [14].

6 Závěr

V rámci možností byl vytvořen kioskový režim a to díky některým vlastnostem přímo v nastavení systému Android. Bylo například zabráněno přechodu do režimu spánku a nastaveno nepřetržité zapnutí displeje tabletu. Dále bylo v průběhu programování a ladění aplikací zjištěno, že pokud by se využívalo i běžného režimu systému Android, je nutné dbát na zmíněné nastavení systému, neboť mohou některé aplikace kioskový režim narušovat. Například zvukovým ohlášením zprávy týkající se aplikace facebook. Může za to způsob, jakým je kioskový režim realizován. Běží totiž jako aplikace v systému Android, která ale nezobrazuje veškeré aplikace v tabletu.

Dále byly vytvořeny tři aplikace, které budou v elektromobilu využity. Jedna přináší řidiči zábavu v podobě MP3 přehrávače, druhá poskytne přehled uživateli o stavu světel a třetí informuje řidiče o stavu vozidla. Aplikace Osvětlení zobrazuje stav světel pomocí dostatečného grafického znázornění, které dodá řidiči potřebný přehled o aktuálním stavu daných světel. Třetí aplikace Informační panel je připravena zobrazit některé informace, týkající se elektromobilu. Patří mezi ně zařazený rychlostní stupeň, rychlost, stav baterie, teplota baterie. Dále též odhadovaný dojezd na baterii, celkový počet najetých kilometrů, měřič najetých kilometrů a zda je odpojené, či připojené dobíjecí zařízení. Stav dobíjecího zařízení je simulováno skrze napájení tabletu, tedy dle připojení nebo odpojení USB kabelu, který je napájen ze zásuvky. V aplikaci jsou také vytvořeny simulační rámce sběrnice CAN, aplikace je schopná tyto rámce zpracovat a následně zobrazit zjištěné hodnoty v UI. Aplikace Informační panel má responzivní design a strukturu, kterou lze dále rozšiřovat.

Operační systém tabletu byl odblokován, čímž se podařilo získat práva superuživatele a je možné v budoucnu využít širší možnosti tabletu. Počítá se s rozšiřováním funkcí pro elektromobil, například o aplikace týkající GPS navigace a mnohých dalších perspektiv, které podpoří další vývoj v této oblasti. Práce představuje užitečný výsledek podporující další směry vývoje aplikací v systému Android a jejich využití ve školním elektromobilu eŠus.

Literatura

[1] martinzmartina. BMW i3: Revoluce městského cestování? (Test). Na: ***REFRESHER.cz | Hudba, Moda, Lifestyle*** [online]. martinzmartina, 2015 [cit. 2016-04-19]. Dostupné z <http://refresher.cz/28039-BMW-i3-Revoluce-mestskeho-cestovani-Test>.

[2] martinzmartina Na: ***REFRESHER.cz | Hudba, Moda, Lifestyle*** [online]. [cit. 21.4.2016]. Dostupné z <https://i.refresher.sk/public/martinzmartina/DSC_0389.jpg>.

[3] DUCHOŇ, Jiří. Na: ***Motor Trend: Car Reviews & News – New Car Prices and Ratings*** [online]. [cit. 21.4.2016]. Dostupné z <http://www.autorevue.cz/getthumbnail.aspx?q=100&height=100000&width=100000&id_file=921318261>.

[4] Nissan International SA, Switzerland. NISSAN LEAF: Jak získat maximum z vašeho vozu LEAF. Na: ***NISSAN LEAF*** [online]. Francie: Nissan International SA, Switzerland, 2015, s. 64 [cit. 2016-04-21]. Dostupné z <http://www.trendpark.cz/images/docs/leaf/manual.pdf>.

[5] ZUNA, Dominik. Elon Musk představil nejlevnější Tesla Model 3 a už trhá rekordy v předobjednávkách. Na: ***Letem světem Applem - Magazín o společnosti Apple a produktech Apple*** [online]. Dominik Zuna, 2016 [cit. 2016-04-21]. Dostupné z <https://www.letemsvetemapplem.eu/2016/04/04/tesla-model-3/>.

[6] OGBAC, Stefan. Na: ***AutoRevue.cz – Auta, testy, novinky, fotografie*** [online]. [cit. 21.4.2016]. Dostupné z <http://www.motortrend.com/uploads/sites/5/2016/03/Tesla-Model-3-interior-1.jpg>.

[7] Android Developers. Manifest.permission. Na: ***Android Developers*** [online]. Android 6.0 r1, 2016 [cit. 2016-04-27]. Dostupné z: <http://developer.android.com/reference/android/Manifest.permission.html#READ_EXTERNAL_STORAGE>.

[8] Android Developers. ActivityInfo. Na: ***Android Developers*** [online]. Android 6.0 r1, 2016 [cit. 2016-04-27]. Dostupné z: <http://developer.android.com/reference/android/content/pm/ActivityInfo.html#LAUNCH_SINGLE_INSTANCE>.

[9] Android Developers. Intent. Na: ***Android Developers*** [online]. Android 6.0 r1, 2016 [cit. 2016-04-27]. Dostupné z: <http://developer.android.com/reference/android/content/Intent.html#CATEGORY_HOME>.

[10] Android Developers. Activity. Na: ***Android Developers*** [online]. Android 6.0 r1, 2016 [cit. 2016-04-27]. Dostupné z: <http://developer.android.com/reference/android/app/Activity.html#onBackPressed%28%29>.

[11] Android Developers. USB Accessory: Manifest and resource file examples. Na: ***Android Developers*** [online]. [cit. 2016-04-27]. Dostupné z: <http://developer.android.com/guide/topics/connectivity/usb/accessory.html>.

[12] Android Developers. Monitoring the Battery Level and Charging State: Monitor Changes in Charging State. Na: ***Android Developers*** [online]. [cit. 2016-04-27]. Dostupné z: <http://developer.android.com/training/monitoring-device-state/battery-monitoring.html>.

[13] KOVÁŘ, Bc. Jan. ***Zobrazovací jednotka pro experimentální elektromobil***. Liberec, 2015. Diplomová práce. Technická univerzita v Liberci. Vedoucí práce Ing. Pavel Jandura

[14] konsolen. Asus ME581CL / Downgrade from x.x.77 and root now possible / worldfirst! Na: ***Android Forum for Mobile Phones, Tablets, Watches & Android App Development - XDA Forums*** [online]. konsolen, 2015 [cit. 2016-04-19]. Dostupné z <http://forum.xda-developers.com/general/help/root-downgrade-asus-me581cl-t3075678>.

A UML Class Diagram Informačního panelu