



Střední průmyslová škola na Proseku
190 00 Praha 9, Novoborská 2

OBOR

**18-20-M/01 INFORMAČNÍ TECHNOLOGIE
(ŠVP: INFORMAČNÍ TECHNOLOGIE)**

MATURITNÍ PROJEKT

TÉMA PRÁCE

RobCon

PROHLÁŠENÍ

Prohlašuji, že jsem svou maturitní práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW, atd.) uvedené v seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 10. 4. 2015

.....

podpis

PODĚKOVÁNÍ

Poděkování patří především mému vedoucímu Jiřímu Šihánovi za jeho vedení a rady během vývoje projektu. Děkuji také ostatním učitelům a zaměstnancům mé školy za zkušenosti, vědomosti a prostředí, které mi bylo vždy k dispozici. Také děkuji své rodině a přátelům za jejich podporu.

ANOTACE

Jméno autora	Jakub Bednář
Název maturitní práce	RobCon
Rozsah práce	Počet stran práce ve formátu (např. 165 s. + 50 s. příloh)
Školní rok vyhotovení	2017/2018
Škola	Střední průmyslová škola na Proseku
Vedoucí práce	Ing. Jiří Šilhán
Využití práce	Práci je možné využít jako ovladač, nebo jako obrazovku pro výpis ze zařízení. Aplikaci je možné využít v celém spektru oborů např. výuka, hobby, servis, komerční účely a další.
Klíčová slova	Ovladač, Konzole, swift, iOS, kontrola, robot, zařízení, Arduino, HM-10, Bluetooth
Anotace	Vzdálená kontrola přístrojů pomocí Bluetooth z iOS zařízení pomocí jednoduché aplikace. Aplikace bude komunikovat s přístroji dle specifického nastavení jednotlivých připojených zařízení.

ANNOTATION

Autor	
Title of graduation work	
Extend	Např. 165 p. + 50 p. of appendices
Academic year	2015
School	Střední průmyslová škola na Proseku
Supervisor	
Application	
Key words	
Annotation	

OBSAH

1. Úvod	9
2. Cíl práce	9
3. Arduino	10
3.1. Arduino UNO	10
3.1.1. ATmega 328p	11
3.1.2. Externí sérové sběrnice SPI a I2C	11
3.2. Arduino Shieldy	12
3.2.1. EasyVR Shield 3	13
3.3. Arduino Moduly	13
3.3.1. Bluetooth module HM-10	13
4. Swift	15
4.1. Historie	15
4.2. Základní syntax Swiftu	16
4.3. Swift Playgrounds	16
4.4. Xcode	16
5. Vývoj pro iOS	18
5.1. Nativní aplikace	18
5.2. Hybridní aplikace	18
5.3. Hybridní jádro aplikace	19
5.3.1. Visual Studio	19
5.3.2. Xamarin Studio	19
5.4. Web aplikace	20
5.4.1. Cordova	20
6. RobCon	21
7. Uživatelská dokumentace	22
7.1. Mapa aplikace	22
7.2. Ikony s popisem jejich funkce	23
7.3. Úvodní stránky	25
7.4. Mouse Droid Controller	25
7.5. Console Controller	26
7.6. Music Select Page	26
7.7. Astro Droid Controller	27
7.8. Instalace aplikace	28

7.8.1.Podmínky pro zdárnou instalaci	28
7.9.Postup instalace	29
8. Technická dokumentace	30
8.1.Rozložení funkcionality “popis jednotlivých souborů”	30
8.1.1.AppDelegate	30
8.1.2.Info.plist.....	30
8.1.3.Main.storeyboard	30
8.1.4.ViewControler	31
8.1.5.deviceViewCell.....	31
8.1.6.mseControler	31
8.1.7.mseSound	31
8.1.8.consoleControler	31
8.2.BleSerial (třída)	31
9. Závěr.....	33
10.Použitá literatura	34
11.Seznam zkratk	35
12.Seznam obrázku	35

1. Úvod

Lidstvo již od počátku hledá cesty pro zjednodušení života. Jedním z největších pomocníků které lidstvo objevilo je bezdrátový přenos. Původní metrové antény jsou dnes nahrazeny obvody, které mají několik milimetrů a stojí zlomek ceny. Díky takto příznivým podmínkám je možné využít bezdrátové technologie takřka pro cokoli. Jednou z nejpoužívanějších možností využití je PAN (Personal area network), což je lokální síť pouze v blízkém okolí vysílače jako například Bluetooth nebo Wi-Fi. Bezdrátové sítě toho typu dnes nalezneme na každém rohu, ať už v restauracích, veřejných institucích nebo domácnostech.

V případě, že máme zařízení které samo vytváří vlastní PAN, například pomocí Bluetoothu, proč se na něj jednoduše nepřipojit bez náročného hledání a nevyužít jeho pravý potenciál. Dnes si tento potenciál uvědomují jak výrobci tak uživatelé a proto vzniká množství produktů, které na základě bezdrátového připojení staví. Příkladem takových zařízení mohou být například bezdrátová sluchátka nebo autíčko s kamerou, která obraz přeposílá do kontrolního zařízení.

2. Cíl práce

Teoretická část práce si klade za cíl vysvětlit jednotlivé principy a pomůcky, které jsou dále použité v samotné praktické práci. Cílem práce je popsat jak obecný pohled na danou problematiku, tak konkrétní části finálního projektu. Práce bude obsahovat kapitolu věnovanou čistě vývoji pro iOS, na který je práce zaměřena, ale i na konkurenční Android. S ohledem na vývoj vytváření programu v poslední době se bude jedna kapitola věnovat multiplatformnímu vývoji.

Další významnou částí bude popis platformy Arduino a odvětví kterých se týká, jako elektronika, mechanika, programování a další.

Jelikož celý vývoj aplikace bude probíhat v jazyce Swift, bude jedna z kapitol věnována tomuto jazyku.

Praktická část práce bude obsahovat popis vytvořené aplikace včetně návodu k použití. Popis jednotlivých částí a symbolů. V technické části práce bude popis fungování aplikace a návod jak si ji přizpůsobit pro specifické účely.

V poslední řadě bude práce obsahovat ukázkou kódu pro Arduino, na kterém lze práci demonstrovat. Tento kód bude psaný v upraveném jazyce pro Arduino - Wring.

3. Arduino

Arduino je jednou z velmi populárních platforem pro studenty. První koncept této platformy vznikl v roce 2005 v Ivree (Itálie). Jeden z tvůrců této platformy Massimo Banzil měl oblíbený bar. Tento bar byl pojmenován podle Italského krále Arduina. A to je důvodem pojmenování celé platformy. Od svého založení je Arduino především studentským projektem, který si získal oblibu ve všech generacích mladých programátorů pro svou jednoduchost spojení HW (hardware, fyzické zařízení PC) a SW (software, data a programy) do jednoho funkčního celku. Nedílnou součástí všech projektů je množství senzorů a kompatibilních periférií.

Základní modely Arduina využívají mikrokontroléry ATMega od firmy Atmel. Mikrokontrolér znamená chip, který v sobě obsahuje všechny části nutné k fungování jako počítač, často jsou označovány jako mikropočítače nebo jednočipové počítače. Programování probíhá na externím PC za pomoci Arduino IDE (Integrated Development Environment, vývojové prostředí). Vývojové prostředí je jednoduchým, ale výkonným nástrojem, který je psaný v Javě. Dnes již není nutné IDE instalovat, je možné jej pustit online v prohlížeči. Programování samotné probíhá v jazyce C nebo C++. Pro zjednodušení programování slouží knihovny Wringu staršího projektu stejného zaměření. Arduino se připojuje za pomoci USB -B kabelu. U vyšších modelů se můžeme setkat s možností připojení přes Bluetooth nebo Wi-Fi.

Od počátku své existence je Arduino a všechny jeho součásti pod open source licencí. Z otevřené licence vyplývá možnost volně si upravovat jednotlivé části zařízení nebo IDE dle vlastních potřeb. Otevřená licence znamená také jednoduchou možnost duplikace a tím i vysokou konkurenceschopnost díky nízkým cenám zařízení dovážených jako klony z Číny. Mezi klony patří například Furduino, Robduino, Freeduino a mnoho dalších. Název Arduino je licencován pouze pro desky vyrobené v Itálii.

Arduino v posledních letech rozvinulo novou řadu desek, které jsou postaveny na procesorech od Intelu řady Galileo. Tyto desky jsou schopny spustit úpravy standardních OS (operating system, operační systém) jako Windows nebo Linux. S deskou jde též pracovat starým způsobem přes Arduino IDE.

3.1.Arduino UNO

Arduino UNO je nejrozšířenější model Arduina díky svému poměru ceny a výkonu. Obsahuje 14 digitálních vstupů/výstupů a 6 analogových vstupů. DPS (deska plošného spoje) je napájena 5V většinou z USB (Universal Serial Bus), ale lze ji napájet i z externího zdroje. Arduino je řízeno mikrokontrolérem ATmega 328p a je možné s ním komunikovat pomocí sériové linky SPI a I2C. Je určeno především začátečníkům, což naznačuje již samotný název. Jednou z výhod je možnost vyjmout samotný mikrokontrolér a samostatně jej použít pouze s dalšími čtyřmi součástkami, což značně šetří místo. Aktuální verze desky je 3 a celé označení je Arduino UNO R3.

3.1.1.ATmega 328p

ATmega 328p je nejpoužívanější mikrokontrolér pro řízení Arduina. V klasické verzi jej nalezneme v Arduino UNO. V SMD (zmenšené) verzi jej nalezneme v Arduino Nano nebo Mini. Čip obsahuje 14 digitálních vstupů/výstupů a 6 analogových vstupů/výstupů. Jedná se o 8 bitový mikrokontrolér s 32 KB flash pamětí. O trvalé ukládání se stará 1 KB EEPROM. MCU (Micro Controller Unit - Mikro řídicí jednotka) je taktována na 20MHz. Pro možnost analogového vstupu je na 6 portech možná PWM, tzv. pulzně šířková modulace. Mikrokontrolér si získal oblibu pro plné fungování od 1,8V a to navíc v podmínkách s teplotním rozsahem od -55°C do 125°C. Takové teploty umožňují bezproblémový provoz jak v běžných podmínkách mrazu, tak v podmínkách vysokých teplot způsobených přímým slunečním působením. Všechny údaje lze nalézt v originální dokumentaci.

3.1.2.Externí sérové sběrnice SPI a I2C

SPI neboli (Serial Peripheral Interface) je spolu s I2C (Inter-Integrated Circuit) nedílnou součástí veškeré dnešní elektroniky a to z několika základních důvodů. Nejzásadnějším faktorem jejich rozšířenosti je jejich jednoduchost implementace do zařízení i jejich následná práce softwarová. HW implementace je snadná díky velkému množství chipů na trhu, které jí zvládají. SW implementace je snadná díky množství knihoven ve všech jazycích. Další výhodou je možnost komunikovat s jedním, nebo více zařízeními najednou. Poslední zmíněnou výhodou je přítomnost hodinového signálu (CLK, clock, hodiny) v komunikaci. CLK je frekvence komunikace, na které spolu zařízení komunikují v případě zmíněných sběrnic. Dále je frekvence posílána po speciálním vodiči označovaným SCK. Sběrnice mají

rozdílné taktování CLK přičemž SPI je standardně taktováno na 10MHz a I2C na 3.4MHz.

SPI sběrnice je propojena s dalšími uzly za pomoci čtyř propojek a to již zmíněné SCK pro přenos hodinové frekvence. Komunikace je obousměrná, tedy full duplex a jednotlivé uzly (zařízení) jsou rozděleny na hlavní uzel Master, který řídí hodinovou frekvenci v komunikaci a ostatní zařízení, která se přizpůsobí Slave. Dále se pro komunikaci používají dva vodiče a to MISO (Master in Slave out) a MOSI (Master out Slave in). Posledním kabelem je SSEL (Slave select), ten určuje se kterým Slave uzlem právě zařízení komunikuje. K bezporuchovosti a jednoduchosti prospívá jednosměrný provoz na každé lince. Nevýhodou popsané sběrnice může být v některých případech pouze jeden Master uzel, z čehož vyplývá kratší vzdálenost připojených zařízení kvůli synchronizaci hodinového času.

I2C je druhou zmíněnou sběrnici. Společná vlastnost je již zmíněný hodinový signál. Rozdíl je však v rozesílání dat, jelikož se jedná pouze o poloduplex (komunikace pouze jedním směrem). V komunikaci je jeden pevně nastavený Master a je pevně nastaven protokol. Komunikace je oproti SPI složitější, ale je možné ji využít na delší vzdálenost díky absenci taktovací propojky a úspoře dvou vodičů.

3.2.Arduino Shieldy

Základní Arduino je samostatně fungující celek, který lze jednoduše rozšířit o množství funkcí pomocí takzvaných Shieldů. To znamená, že můžeme nasadit na Arduino další desku, která rozšíří jeho možnosti o internet, hudbu, nebo další funkce podle typu. Základní a nejpoužívanější Shieldy jsou například Ethernet Shield, Music Shield, Motor Control Shield a další. Různé Shieldy můžeme specificky vrstvit podle jejich portů pro komunikaci s Arduinem. Po instalaci desek je nutné upravit kód, což spočívá v nalezení a vložení knihovny do IDE, která většinou obsahuje i vzorové kódy pro ukázkou. Na základě již existujících ukázkových programů je možné sestavit vlastní program, který bude sestaven dle přání programátora. Shieldy se v základu rozdělují na originální, od samotného Arduina, a shieldy třetích stran a kopií z Číny. Cena originálních desek je jejich největší nevýhodou, jelikož v porovnání s klony není většinou v kvalitě a funkčnosti znatelný rozdíl. Existuje řada nezávislých projektů, které přichází s odvážnějšími návrhy vlastních desek, jako například společnost VeeR, mezi jejíž shieldy patří například EasyVR Shield 3. Fungování tohoto shieldu je popsáno v následujícím odstavci.

3.2.1.EasyVR Shield 3

Jedná se o jeden z nejzajímavějších modulů na trhu a to díky svým vlastnostem hlasové komunikace s člověkem. Modul obsahuje mikrofón a reproduktor, díky kterým poslouchá, co člověk říká a v případě shody s uloženými zvukovými záznamy je schopen obratem pustit přednastavený zvuk do reproduktoru. Tímto postupem je možné navodit u uživateli pocit lidské interakce se zařízením, což je v dnešní době velice žádaná a probíraná problematika. Kromě pouhé odpovědi pošle Shield informaci o přijaté i odeslané zvukové stopě po sériové lince do Arduina. Program v Arduinu může po obdržení Shieldem odeslaných dat vyvolat další reakce, které slouží k vyvolání vnější reakce. Modul je možné nalézt ve třech generacích, přičemž aktuální je právě generace 3. Každá generace modulu se prodávala ve dvou verzích. Jedna verze je v prodeji jako již postavený modul, který je stavěn přímo na Arduino. Druhá verze obsahuje modul, který je vhodný především do vlastních projektů, kde je třeba brát ohledy na prostor. Modul je možné připojit k jakékoli vývojové platformě jako například Raspberry Pi, Banana Pi, nebo k jakémukoli zařízení se Sériovou linkou.

3.3.Arduino Moduly

Arduino je také možné rozšířit o různá I/O (vstupní/výstupní) zařízení, jako například senzor zvuku, senzor světla, senzor pohybu, LCD display (liquid crystal display, displej z tekutých krystalů), diody, reproduktor a další. Sensory je možné si postavit, nebo objednat již zkonstruované. Díky rozsáhlé komunitě je na internetu zdarma ke stažení množství schémat. V případě zajmu naprostého začátečníka je možné nalézt množství celých tutorialů, kde je celý popis výroby a následného zapojení. Výroba vlastních senzorů je však relativně nákladná a náročná oproti nákupu z velkých globálních e-shopů jako aliexpress.com, ebay.com nebo amazon.com.

3.3.1.Bluetooth module HM-10

Jedná se o malý modul s šesti vývody, přičemž pro komunikaci jsou nutné čtyři a to Vcc, GND, TX, RX (napájení +, napájení -, vysílání, přijímání). Oproti nižším

modelům jako například HC-5 nebo HC-6 je tento modul vystaven na novějších standardech a to Bluetooth 4.0 BLE (Bluetooth Low Energy), díky čemuž je možné jej použít i s novými zařízeními, která z energetických důvodů komunikaci na starším standardu nedovolovala. Modul komunikuje s Arduinem, nebo jinou řídící platformou pomocí AT příkazů, které posílá po sériové lince. Modul je řízen chipem CC2541, který obstarává veškeré fungování komunikace a to jak na stranu přijímání z připojeného zařízení, tak na stranu odesílání do řídícího obvodu. V běžných podmínkách je dosah vysílače/přijímače kolem 15 metrů. Teoreticky se však udává až 60 metrů a to při napájení standardními pěti volty. Modul je možné nakonfigurovat i pro samostatné fungování, kdy nepotřebuje žádnou řídící jednotku, jelikož jí dokáže být sám. Další konfigurací lze dosáhnout nahrávání programu z pc do Arduina právě přes tento modul.

4. Swift

Swift je programovací jazyk vytvořený Chrisem Lattnem a Apple Inc. Jedná se o multiparadigmatický programovací jazyk, což znamená, že v něm lze pracovat ve více programovacích paradigmatech, způsobech. Swift například umožňuje přistupovat pomocí Objective C ke knihovně napsané v C++ a přitom programátor ovládá uživatelské rozhraní přímo za pomoci Swiftu. To je možné uplatnit na všech platformách Applu, tedy na iOS, macOS, watchOS a tvOS. Je možné jej využít i pro programování pro Linuxové platformy nebo jiné.

Zařazuje se do rodiny vyšších programovacích kompilovaných jazyků. Jejich základní vlastností je, že před spuštěním je nutné je převést do nižšího jazyka (strojového kódu) pro spuštění aplikace. Do verze 2.2 se jednalo o proprietární jazyk, od té doby je pod open-source licenci. Primárním IDE je Xcode vyvíjený Apple, který jej implementoval ve verzi 6. Koncept jazyka je vystaven na Objective C. Jednou z výhod je, že může pracovat s frameworky a knihovnami Objective C, ale i C a nemá problém se zpracováním souboru obsahujících C++. Jazyk je úzce spojen s platformou Cocoa a Cocoa Touch framework, která slouží k vytvoření UI (user interface, uživatelské prostředí). Aktuální verze jazyka je 4.0. Verze 4.0 byla představena v roce 2017 jako již tradičně na WWDC (Apple Worldwide Developers Conference).

4.1.Historie

Vývoj jazyka začal v roce 2010 Chris Lattner. V průběhu vývoje se k němu připojil Apple se svým týmem. Během vývoje jazyka bylo k základu postaveném na Objective C přidáno množství vlastností jiných jazyků jako například C#, Python, Rust a další. Díky přebírání výhod různých jazyků bylo dosaženo jednoho z nejkomfortnějších a nejnovějších jazyků dneška. Poprvé byl Swift otevřeně představen v roce 2014 na WWDC. Již při prvním představení byla vydána dokumentace, která čítala 500 stran. Pro lepší dostupnost byl manuál v knižní podobě přidán do iBooks (služba Applu na prodej e-knih). Tato verze byla dostupná pouze pro testování a nedočkala se užití pro aplikace v AppStore.

Druhá verze s označením Swift 2 byla představena na WWDC 2015. Její nejpodstatnější novinkou byla možnost zveřejňovat své aplikace do obchodu. Díky

možnému budoucímu využití jak pro komerční tak pro nekomerční účely se již Swift stal plnohodnotným hráčem na vývojovém trhu aplikací.

Třetí a čtvrtá verze přináší vylepšení stability a kompatibility. Dnes jsou v iBooks dostupné nejen dokumentace verzí, ale i návody a příručky pro začátečníky i pokročilé přímo od firmy Apple, nebo od vývojářů třetích stran.

4.2.Základní syntax Swiftu

Základní kód Swiftu je lehce srozumitelný. Proměnné jsou rozdělené na var , což jsou proměnné jako takové a let, což jsou konstanty. K jednoduchosti přispívá automatické doplňování datových typů podle vložené hodnoty. Všechny druhy operátorů jsou stejné jako v C nebo PHP.

4.3.Swift Playgrounds

Jedná se o aplikaci vyvíjenou firmou Apple od roku 2016. Aplikace je určena především dětem, které se díky ní můžou intuitivní formou učit programovat právě ve Swiftu. Aplikace je dostupná pouze pro iPady. Učení probíhá plněním úkolu, přičemž program studentům ukazuje nápovědu k právě nutným krokům. Výsledek se ukazuje na druhé půlce obrazovky. Pro názornost je možné programovat i různé druhy inteligentních vzdělávacích hraček. Jednou z nejrozšířenějších “hraček” je LEGO® MINDSTORMS® EV3, což je inteligentní rozšíření kostek LEGO® Technik, které lze kombinovat s klasickými LEGO® kostkami. Je možné programovat i další druhy robotů, dronů a vozítek. V novějších verzích aplikace přináší podporu rozšířené reality, což zasazuje interaktivní výuku do reálného světa.

4.4.Xcode

Xcode je IDE pod správou Applu určené výhradně pro zařízení s macOS. Program byl představen veřejnosti v roce 2003, kdy byl určen především pro programování v Objective C. Dnes prostředí nativně podporuje C, C++ (knihovny), Objective-C, Objective-C++, Java, AppleScript, Python, Ruby, ResEdit (Rez), a Swift. Prostředí je určeno pro vývoj na zařízení od Applu jako například iPhone, Mac, Apple

Watch a další. Je možné vyvíjet jednoduché aplikace nebo komplexní hry. Nástroj integruje mnoho pomůcek a postupů k vývoji. Je možné si zvolit, zda aplikaci psát ve Swiftu, nebo použít Metal framework (grafická knihovna), či si zvolit jinou cestu. Jednou z největších výhod je takzvaný Interface Builder, který pomůže vývojáři jednoduše seskládat UI jeho aplikace bez použití značkovacích jazyků. Oproti většině ostatních programů má výhodu v integrovaném simulátoru výsledných zařízení, které poskytují komplexnější zpětnou vazbu při vývoji, než standardně používané emulátory. Po doinstalování daných frameworků je možné Xcode využít i ke konzolovému programování řady jiných jazyků jako například C, Python, nebo prostředí upravit pro psaní a nahrávání programů do Arduina.

5. Vývoj pro iOS

V minulosti byl vývoj pro iOS možný pouze v Xcode, kterou poskytoval sám Apple. Problémem bylo, že Xcode byl a stále je dostupný pouze pro macOS. Situace ohledně vývoje se změnila a dnes je možné využít řadu nástrojů, přístupů nebo jiných možností jak dosáhnout vlastní aplikace na jablečných zařízeních. V základu se vývoj rozdělil na nativní a hybridní aplikace. Ty se dále rozdělují na webové aplikace a aplikace se sdíleným jádrem. Každá z těchto metod vývoje má své výhody, ale i svá úskalí a proto je důležité vědět, kdy, jak a kterou metodu využít. Ačkoli je snahou všech vývojářů jednotlivých přístupů dosáhnout co nejsnazšího vývoje a zároveň nejlepší funkčnosti jejich nástrojů. Často dochází k chybám a zbytečným defektům.

5.1.Nativní aplikace

Nativní aplikace jsou aplikace vytvořené přímo pro určitou platformu, což znamená, že aplikace je psána na míru danému systému. V případě iOS je nejvíce nativních aplikací napsaných v již zmíněném Xcode. Existují však i alternativy, které umožňují vyvíjet iOS aplikace i na Windows nebo Linux. Jedním z nejpopulárnějších prostředí tohoto druhu je Visual Studio od Microsoftu. Výhodou tohoto prostředí je psaní zdrojových kódů v C# a využití XAML pro vytváření grafiky. Prostedí je vystavěné na Xamarin Studiu, které bylo jeho předchůdcem a je stále podporované. Jedná se však o stejný program s jiným jménem taktéž pod zprávou Microsoftu. Dalším nástrojem pro vývoj je například Eclipse nebo Swifty.

5.2.Hybridní aplikace

Všechny hybridní aplikace jsou aplikace od začátku plánované na multiplatformní využití. Multiplatformním využitím se myslí chod jedné aplikace napříč platformami. Rozdílnost systémů je však značná a proto psaní multipatformně není vždy dokonale možné a je třeba najít klíčku, kterou dosáhneme správné funkce. Dnes jsou nejrozšířenější dva způsoby jak dosáhnout multiplatformnosti s překonáním těchto překážek. Prvním způsobem je psaní obecné aplikace za pomoci multiplatformních frameworků, které obsahují funkcionalitu všech podporovaných

OS. Druhým řešením je psaní aplikace jako webové stránky, která se zobrazuje v zařízení jako aplikace, i když se jedná pouze o web.

5.3.Hybridní jádro aplikace

Aplikace obsahující tzv. sdílený kód jsou jedním z neoptimálnějších multiplatformních přístupů dneška. Ve své podstatě se jedná o napsání aplikace v univerzálním jazyce pro prostředí a následné zkompilování pro určitou platformu. Pro využití specifických funkcí zařízení jako jsou Bluetooth či databáze, se využívají frameworky, které jsou taktéž stavěny multiplatformně. Na sdíleném jádru je možné vytvářet multiplatformní UI, nebo jej vytvořit pro každou platformu specificky. Hojně využívaným nástrojem tohoto typu je Xamarin Studio nebo Visual Studio.

5.3.1.Visual Studio

Jedním ze softwarů tohoto typu je Visual Studio Mac (Preview). Nástroj je v této verzi dostupný pouze na macOS. Jedná se již o stabilní a používanou verzi. Tento výkonný nástroj umožňuje vývoj aplikací na macOS, iOS a Android. Je zde taktéž možné psát konzolové aplikace. Nástroj obsahuje jednoduchého manažera pro zprávu nuget. Díky tomuto rozšíření je zde velice jednoduché přidávat, spravovat nebo odstraňovat jednotlivé frameworky třetích stran. Dříve bylo Visual studio výhradou pouze Windows. S rokem 2016 se však objevila verze pro macOS, která je postavena na Mono frameworku. Tento doplněk umožňuje spuštění C# aplikací na jiných platformách než Windows.

5.3.2.Xamarin Studio

Dalším původnějším nástrojem je Xamarin Studio Community. IDE bylo představeno v roce 2013, kdy ještě bylo placené a na všech platformách. Dnes jej nalezneme pouze na macOS. Pro Windows je ke stažení pouze jako plugin do Visual Studia. Xamarin obsahuje mnoho možností k vývoji. Pro vývoj UI je zde možné použít XAML preview, které živě kompiluje XAML kód a zobrazuje jej na druhou část obrazovky. Pro snadný přístup k iOS designu je možné využít Apple Storeboard, což je jednoduché "klekátko" pro vytvoření UI. Ke své funkci využívá Mono frameworky, díky kterým dokáže kompilovat C# na Macu. Obsahuje také velké množství dalších modulů pro usnadnění práce jako: Debugger, TestCloud, Help, Xamarin Forms

(nástroj pro multiplatformní vývoj). V Xamarinu vyvíjí své aplikace například 3M, Honeywell, The World Bank a další globální a lokální společnosti.

5.4.Web aplikace

V tomto případě vývoje se nejedná přímo o aplikaci. Aplikace, která vznikne tímto způsobem, je pouze ikonka a webový prohlížeč. Obsah aplikace je webová stránka, na které veškerou funkcionalitu zajišťuje JavaScript. Vývoj tímto směrem je všemi směry nejjednodušší. Bohužel využití všech možností zařízení není možné. Proto je dobré si vždy vybrat správnou cestu vývoje na základě výsledného díla. Nejpoužívanějším nástrojem pro vytvoření takovéto aplikace je Cordova, do které vložíme webovou stránku a ona automaticky vygeneruje výsledné aplikace pro vybranou platformu/platformy.

5.4.1.Cordova

Apache Cordova je mobilní framework pro multiplatformní vývoj aplikací. Framework byl představen v roce 2011 s názvem PhoneGap. Později se z původně uzavřeného frameworku stal open source s názvem Apache Cordova. Za pomoci nástrojů HTML5 a CSS3 a JavaScriptu, které jsou v různých platformách již vestavěné, je možné jejich širokospektré základní využití. V pozdějších verzích se do aplikací začaly dávat části nativního kódu, který přidává na funkcionalitě, které by prostá webová stránka nebyla schopna. Mezi tyto funkce například patří přístup k hardwarovým součástem zařízení jako například NFC (Near field communication) nebo Bluetooth. Cordova zvládá pouze jednodušší aplikace, ale zvládne je aktuálně zkompileovat pro 8 platforem, mezi které patří iOS, Android, Windows, LGwebOS, FirefoxOS, Ubuntu, Blackberry10 a další.

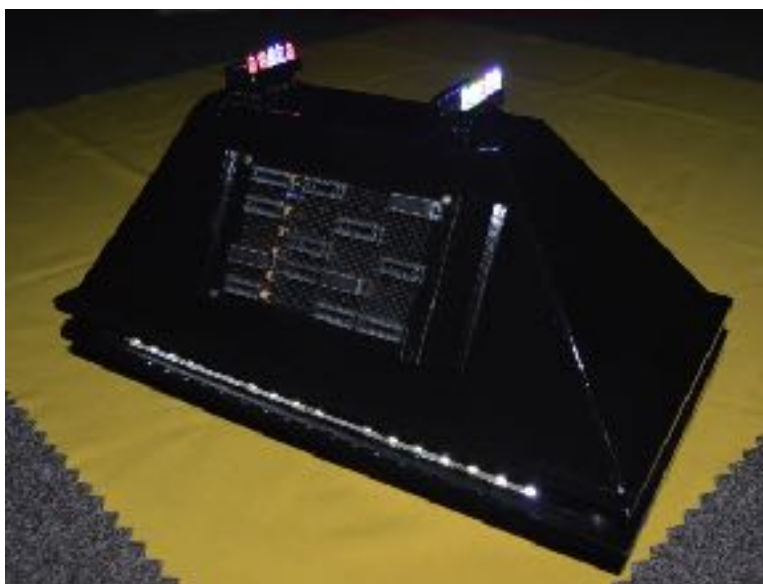
6. RobCon

RobCon je odvozeno od "Robot" a "Control" neboli "robot" a "ovládání" z čehož je možné vyvodit použití aplikace. Aplikace je vlastně jednoduchý nástroj pro moderní způsob ovládání zařízení na dálku. Zařízení jako např. sériový modul "HM-10". Možnosti uplatnění jsou značné díky základnímu zobrazení konzole, která je obecně považovaná za původní nástroj pro kontrolu a zprávu zařízení. Logo aplikace je na Obrázku 1: Logo RobConu.



Obrázek 1: Logo RobConu

Pro testování byl postaven robot přibližně velký 40x23cm a 30 cm vysoký. Srdcem robota je microcontroller AtMega 328p a Bluetooth modul HM-10. Pohon robota je postaven na bázi tanku, což znamená, že má dva motory a jedno volné otočné kolečko. Na vrchu je robot osazen diodami, které svítí "blikají" podle právě hrající hudby. Celkově má robot černou barvu a tvarem připomíná rakev. Po obvodu robota je zabudovaný led pásek, který tvoří jeho osvětlení, taktéž se po obvodu nacházejí senzory vzdálenosti, které zabraňují srážce. Na zadní spodní straně je umístěno světlo pro případného chodce za robotem. Program robota je psán v jazyce Wring. Podoba robota je znázorněna na Obrázku 2: Testovací robot MSE.

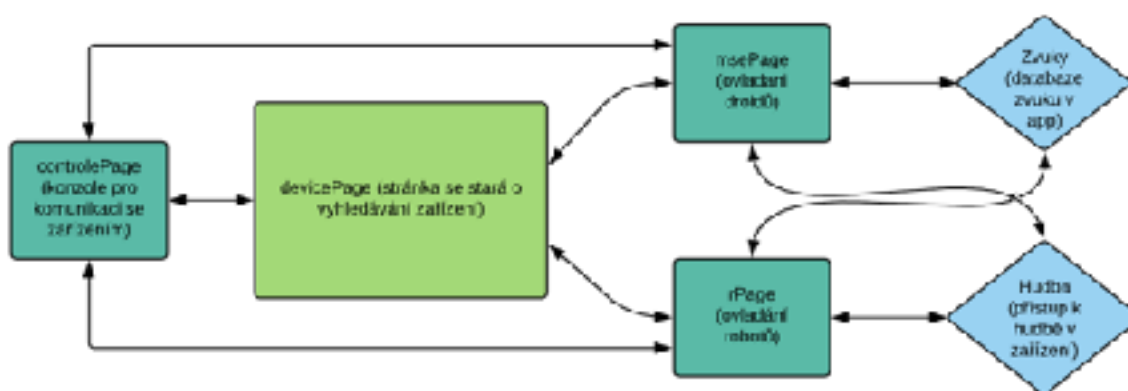


Obrázek 2: Testovací robot MSE

7. Uživatelská dokumentace

7.1.Mapa aplikace

Na Obrázku 3: Mapa aplikace: je znázorněna posloupnost jednotlivých stránek v aplikaci, pomocí diagramu. V diagramu je popsána obecná funkce stránky. Konkrétní funkce jsou popsány v příslušných následujících odstavcích. Obousměrná šipka znamená možnost navrátit se ze stránky zpět na stránku předchozí, nebo stránku znovu vyvolat.



Obrázek 3: Mapa aplikace

Vyvolání kontroléru je závislé na názvu BLE modulu v zařízení. Což znamená že pokud se zařízení bude jmenovat **"MSE-6"** automaticky se jako ovladač otevře **msePage** (Mause Droid Controler). V případě že chceme vyvolat **rPage** (Astro Droid Controler), název zařízení bude **"R2-D2"**. V případě **jakéhokoliv jiného názvu** se vyvolá konzole (**Console Controler**) pro její univerzálnost. Pro případ potřeby změny je možné změnit Controler, nebo název na který se má Controler otevřít v "Config.swift".

7.2. Ikony s popisem jejich funkce

Každá ikona v aplikaci je určena k jedné specifické akci, proto je možné zobecnit funkce jednotlivých tlačítek, které tyto ikony charakterizují. V následujících odstavcích je vysvětlena každá ikona (tlačítko) dle výkladu použitého v aplikaci. Všechny ikony jsou pod GNU GPL. Písmenko v (závorce) znamená příkaz zaslaný zařízením. V případě dvou znaků se jedná o negující prefix, který akci ukončí.

Action



- překlad - akce
- význam - představuje nějaký proces nebo akci
- akce - tlačítko je specifické pro svou multifunkčnost, může na něj být navázána jakákoliv akce dle nastavení robota (u)

Arrow Down



- překlad - šipka dolů
- význam - představuje směr pohybu
- akce - uvádí motor do zpětného chodu (d, l), (bd, bl)

Arrow Up



- překlad - šipka nahoru
- význam - představuje směr pohybu
- akce - uvádí motor do chodu v přirozeném směru (n, p), (bn, bp)

Back



- překlad - zpět
- význam - představuje pohyb hudebních stop zpět
- akce - přetočí aktuálně přehrávanou stopu zpět

Console



- překlad - konzole
- význam - představuje konzoli
- akce - symbol konzole uživatele přesměruje na stránku, která poskytuje konzolovou komunikaci

Lamp



- překlad - lampa
- význam - představuje cílené světlo
- akce - rozsvítí světlo, určené k osvětlení cesty uživateli (a)

Light



- překlad - světlo
- význam - představuje osvětlení robota a jeho okolí
- akce - rozsvítí světlo, které je určeno k osvětlení robota a jeho

okolí, především pro efekt (s)

**Loud**

- překlad - hlasitý
- význam - představuje zvuky nikoli hudbu
- akce - otevře stránku se zvuky, které aplikace obsahuje nebo v případě výpisu je pustí

**Mouse**

- překlad - myš
- význam - představuje druh robota (Mouse Droid)
- akce - přesměrování na kontrolér pro tento druh robota

**Music**

- překlad - hudba
- význam - představuje hudbu
- akce - zobrazí prohlížeč hudby v zařízení

**Next**

- překlad - další
- význam - představuje pohyb skladeb ve frontě přehrávače
- akce - přetočí aktuálně přehrávanou stopu

**Pause/Play**

- překlad - stopni/pust'
- význam - představuje samotný hudební přehrávač
- akce - v případě hrající hudby ji zastaví a opačně

**R2**

- překlad - "není"
- význam - představuje další druh robota s více proměnnými
- akce - přesměrování na kontrolér pro tento druh robota

**Arm Down**

- překlad - ruka dolů
- význam - představuje pohyb ruky robota
- akce - stáhne ruku robota zpět k němu (y)

**Rotate Arm**

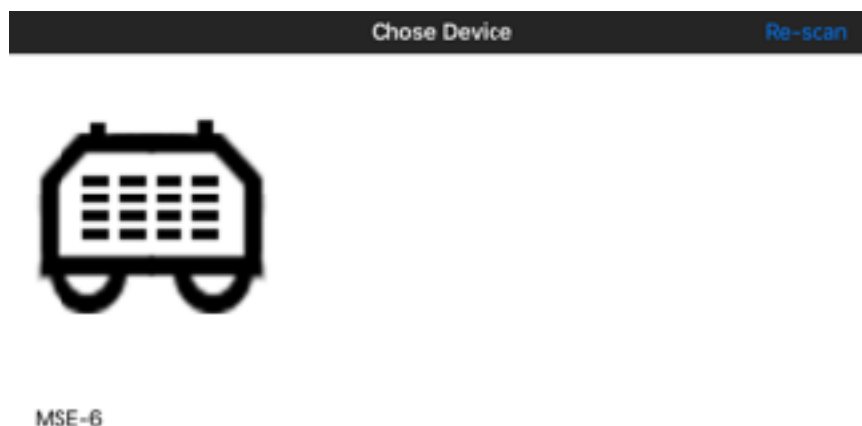
- překlad - otáčet rukou
- význam - představuje pohyb ruky robota
- akce - ruka robota se začne otáčet (c)

**Arm up**

- překlad - ruka vzhůru
- význam - představuje pohyb ruky robota
- akce - zvedne ruku robota do aktivní polohy (x)

7.3.Úvodní stránky

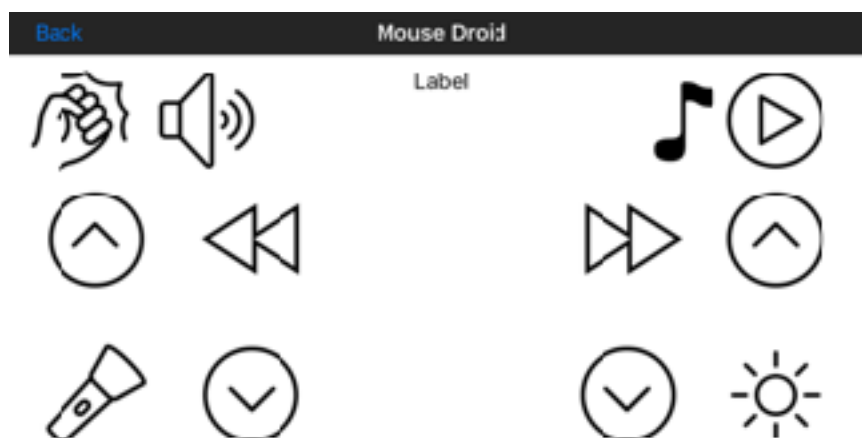
První stránka která se otevře po otevření aplikace je prázdná stránka s tlačítkem “Re-scan” v pravém horním rohu. Tlačítko slouží k vyhledání zařízení v okolí. Po nalezení zařízení se dle jeho typu vypíše na obrazovce. Po vypsání zařízení je možné na zařízení kliknout a tím se k němu připojit. Ukázka úvodní stránky je na Obrázku 4: Úvodní stránka



Obrázek 4: Úvodní stránka

7.4.Mause Droid Controler

Stránka určená k ovládání zařízení typu mause droid. Stránka se skládá z ovládacích prvků, které jsou popsány v kapitole 8.2 Ikony s popisem jejich funkcí. Prázdnou část uprostřed stránky vyplňuje prostor pro výpis logu ze zařízení. “Label” nad prázdnou částí je prostorem pro výpis názvu právě hrající skladby. Ukázku stránky nalezneme na Obrázku 5: Mause Droid Controler



Obrázek 5: Mause Droid Controler

7.5.Console Controler

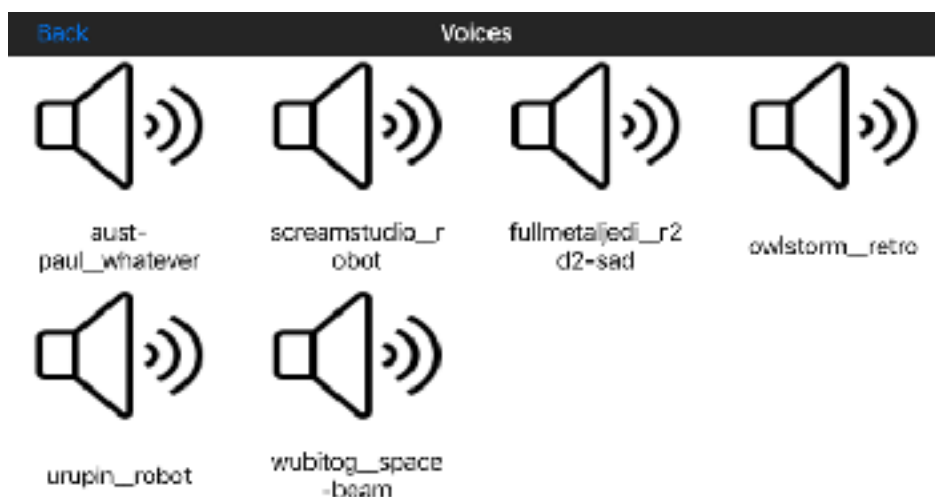
Nejuniverzálnější stránka projektu sloužící k základní komunikaci se zařízením. V horní části je pole pro vepsání příkazu, který lze odeslat do zařízení pomocí "Send" v pravém horním rohu. Console Controler je vychozí stránka, která se otevírá zařízením s nedefinovaným názvem. Konzoly nalezneme na Obrázku 6: Console Controler



Obrázek 6: Console Controler

7.6.Music Select Page

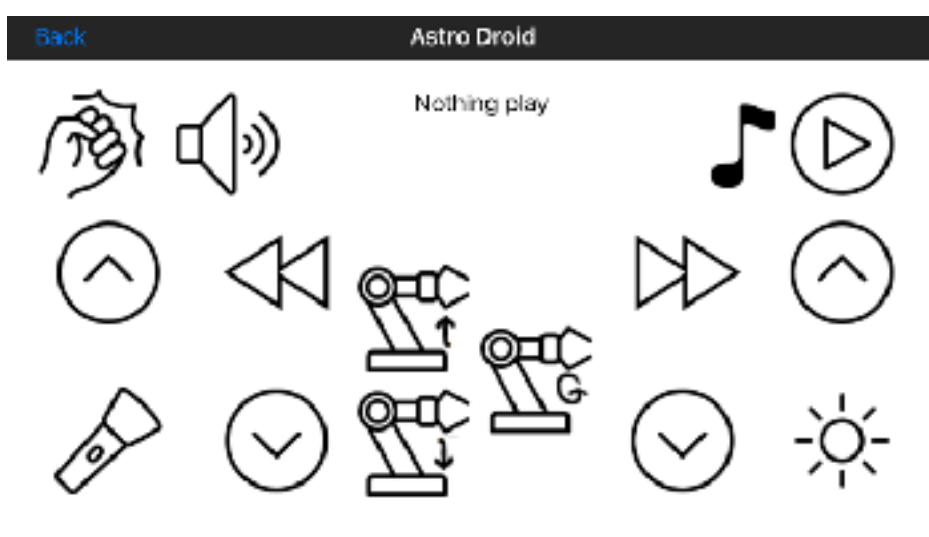
Stránka s výpisem zvuku obsažených v aplikaci. Po vybrání zvuku se automaticky začne přehrávat na pozadí, což znamená, že může odejít ze stránky a zvuk dohraje bez přerušení. Ukázka výpisu zvuku je na Obrázku 7: Stránka výpisu zvuků



Obrázek 7: Stránka výpisu zvuků

7.7.Astro Droid Controler

Stránka určená k ovládání robotů, kteří jsou osazeni otočnou rukou. Stránka se skládá z ovládacích prvků, které jsou popsány v kapitole 8.2 Ikony s popisem jejich funkcí. “Nothing play” nad prázdnou částí je prostorem pro výpis názvu právě hrající skladby. Prázdnou část pod “Nothing play” vyplňuje prostor pro výpis logu ze zařízení. Pod touto prázdnou částí se nachází prvky pro ovládání ruky robota. Ukázku stránky nalezneme na Obrázku 8: Astro Droid Controller



Obrázek 8: Astro Droid Controller

7.8.Instalace aplikace

Aplikace je psána na macOS v Xcode a je určená pro iPhone s iOS 9.3 nebo novější. Pro zdárnou instalaci je třeba splnit několik základních podmínek, které jsou vypsány v kapitole 7.8.1 Podmínky pro zdárnou instalaci.

7.8.1.Podmínky pro zdárnou instalaci

Podmínky pro Mac

- Nejnovější macOS
- Nejnovější aktualizace Xcode
- Přístup k internetu
- Připojený iPhone k Mac pomocí datového kabelu
- Samotná aplikace
- ověřený účet Apple Developer (stačí pouze účet bez zakoupení licence) účtem je možné vygenerovat nový certifikát

Podmínky pro iPhone

- iOS 9.3 nebo novější
- odkliknuté “**Důvěřovat** tomuto zařízení”
- App je optimalizovaná pro iPhone 5 a novější včetně iPhone X

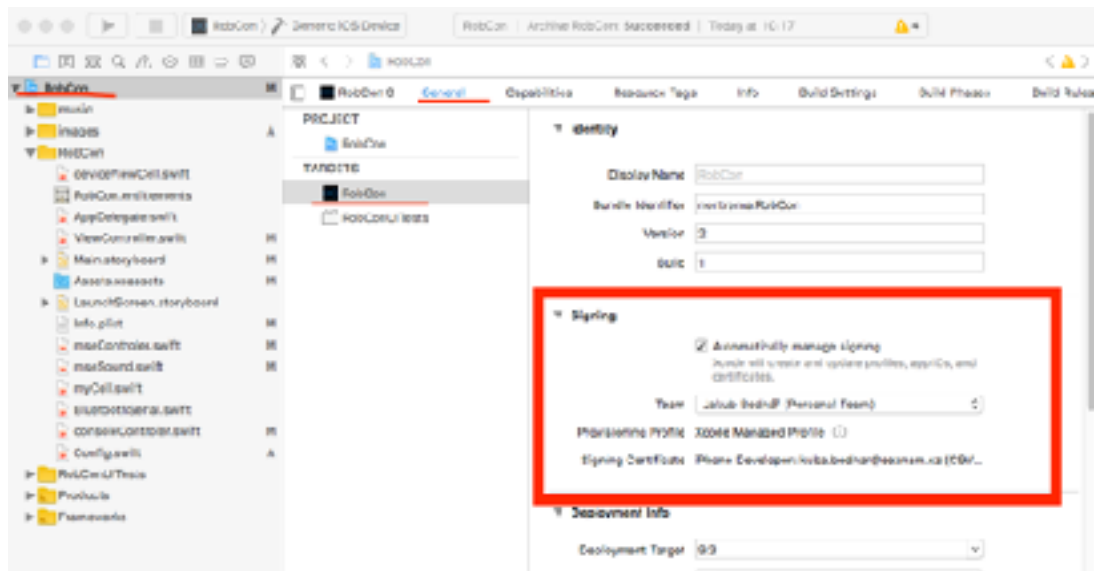
Apple Developer účet opravňuje uživatele licencovat aplikaci pro iPhone s omezením, že přidělený certifikát bude platný pouze 7 dní včetně dne nainstalování. Po uplynutí této doby, nebude aplikace funkční do přehrání s novým certifikátem.

Výchozím jazykem při instalaci aplikace je angličtina, v případě instalace na zařízení, které je lokalizované do češtiny se jazyk automaticky přepne do češtiny.


7.9.Postup instalace

V průběhu instalace je možné, že bude třeba zadat heslo k uživatelskému účtu na Mac.

1. Na Macu otevřeme staženou aplikaci z GitHubu (<https://github.com/Legiani/RobCon>) pomocí Xcode
2. Po otevření zkontrolujeme platnost certifikátu. Příklad platného certifikátu nalezneme na Obrázku 9: Platný certifikát



Obrázek 9: Platný certifikát

3. V případě neplatnosti certifikátu je nutné přidat vlastní Apple developer účet na který se vygeneruje nový certifikát.
4. Pro nahrání do zařízení je nutné v levém horním rohu nastavit namísto "Generic iOS Device" nebo simulátoru, vlastní připojené zařízení např. "Kuba-iPhone".
5. Po nastavení zařízení, do kterého bude aplikace nahrána je nutné spustit kompilaci. Tu provedeme v levém horním rohu tlačítkem pro spuštění (Play) 
6. Xcode by měl začít kompilovat aplikaci. V případě chyby "signal SIGABRT" je nutné vyčistit řešení aneb (Product -> Clean).
7. V případě že iPhone není odemčený je nutné jej odemknout, jinak na tuto skutečnost Xcode upozorní.
8. Aplikace je nahrána. Ale je možné, že aplikace bude požadovat ověření vývojáře, které je nutné na iPhoneu nastavit (Nastavení -> Obecné -> Správa zařízení -> e-mail vývojáře -> "Důvěřovat tomuto vývojáři").
9. Aplikace by měla být spustitelná.

8. Technická dokumentace

Aplikace je psána ve Swiftu a slouží k bezdrátové Bluetooth komunikaci s vnějšími zařízeními. Jedná se o druhou verzi této aplikace. V první verzi bylo pouze ovládání motoru robota. V aktuální verzi je tento základ postaven na čistém kódu a vyšší funkcionalitě.

UI (user interface) neboli uživatelské prostředí je vytvořené pomocí Storeyboardu, což je nástroj k tomu určený. Každá proměnlivá část ve storeyboardu musí mít předdefinovanou třídu pomocí které se s ní pracuje. Jednotlivé třídy jsou obecně popsány v další kapitole 8.1. Rozložení funkcionality “popis jednotlivých souborů”. Přesný popis jednotlivých funkcí a příkazů v aplikaci je popsán přímo v kódu aplikace.

Některá funkčnost může být změněna přímo v aplikaci, nebo v Config.swift který obsahuje definování některých zásadních proměnných pro aplikaci jako například vypisované zvuky, příkazy pro komunikaci, otevření Controleru dle názvu BLE.

8.1.Rozložení funkcionality “popis jednotlivých souborů”

8.1.1.AppDelegate

Jedná se o delegáta aplikace. Delegát je sada instrukcí, případně dat, která jsou uložena mimo aktuální okna a jsou používána bez nutnosti dědění při každém přechodu na jinou stránku.

Stará se o životní cyklus aplikace. Mezi jeho schopnosti můžeme řadit kontrolu, zda aplikace není zaseklá či špatně zkompilevaná. V případě problému s pamětí se o problém stará taktéž AppDelegate.

8.1.2.Info.plist

Jedná se konfigurační soubor aplikace. Jelikož aplikace využívá přístup k hudební knihovně, v telefonu a bluetooth bylo nutné do Info.plist dopsat patřičná oprávnění.

8.1.3.Main.storeyboard

Hlavní soubor který obsahuje stránky a prvky na nich. Neobsahuje však jen jednotlivé prvky, ale i definice přidružení jednotlivých prvků k třídám.

Po přidružení třídy k oknu aplikace je možné ve třídě definovat a pracovat s jednotlivými prvky na stránce.

8.1.4.ViewControler

Třída definovaná k obsluze stránky s výpisem vyhledaných zařízení. Třída využívá BleSerial.swift pro práci s bluetooth. Z vyhledaných zařízení s charakteristickým identifikátorem ve třídě BluetoothSerial označovaným UUID. Je vytvořen výpis zařízení pro uživatele.

8.1.5.deviceViewCell

Třída určená k přístupu k Cell (okénko s zařízením). Ve třídě se definuje přístup k prvkům ze storeyboardu.

8.1.6.mseControler

Třída určená k obsluze stránky Mause Droid Controler, která je popsána v kapitole 7.4 Mause Droid Controler. Třída definuje komplet reakcí na jednotlivé prvky na obrazovce. Jednotlivé příkazy jako reakce na ovládací prvku jsou definované v Config.swift.

8.1.7.mseSound

Třída obsahuje výpis všech zvuků, které obsahuje aplikace. Zvuky které jsou v aplikaci obsaženy jsou pod GPL-GNU licencí. Pole zvuku je vypsáno pomocí třídy "myCell", která definuje přístup k jednotlivým Cell (okénkům výpisu) dle sestavení třídy ve storeyboardu.

8.1.8.consoleControler

Třída určená k obsluze stránky Console Controler která je popsána v kapitole 7.5 Console Controler. Třída přeposílá uživatelem zadané příkazy do zařízení a příkazy ze zařízení zpět.

8.2.BleSerial (třída)

Jedná se o přepracovanou třídu od uživatele GitHubu s přezdívkou hoiberg. Originální třídu můžeme nalézt zde: <https://github.com/hoiberg/swiftBluetoothSerial>. Jedná se o pomocnou třídu k usnadnění práce s Bluetooth na iOS zařízeních, která je určená přímo pro komunikaci s modulem HM-10, nebo jeho vyšší variantu HM-11.

Modul je podrobněji popsán v kapitole 3.3.1 Bluetooth module HM-10. Třída je psána ve Swiftu 4 díky čemuž je její implementace velice jednoduchá. Po přidání třídy do projektu je nutné v každé třídě, kde jí chceme využít, třídu inicializovat.

Autor originální třídy jí uvádí pod MIT Licencí. Podmínky dovolují třídu libovolně používat, **kopírovat**, upravovat, sloučit, publikovat nebo distribuovat. Dle pravidel uvedené licence taktéž autor uvádí, že není zodpovědný za jakékoliv škody v případě chyby v kódu. Kompletní licenční soubor je k nahlédnutí zde: <https://github.com/hoiberg/swiftBluetoothSerial/blob/master/LICENSE>.

Jednotlivé metody třídy jsou popsány přímo ve zmíněné třídě, která je součástí přiložené aplikace (RobCon).

9. Závěr

V úvodu práce je vysvětlena problematika ze které následně vychází celá práce ať se jedná o arduino které je v ovládaných zařízení použito jako řídicí jednotka nebo modul HM-10 který je určen ke komunikaci. V úvodních kapitolách jsem taktéž shrnul způsob komunikace jednotlivých zařízení které jsou nutné k otestování aplikace. Úvod práce taktéž obsahuje teorii která je nutná k osvětlení problematiky programování pro samotnou práci, ale i aktuální trendy ostatních případně multiplatformních přístupu k programování. Cílem práce bylo vytvořit aplikaci která bude schopná obsluhovat roboty případně vypisovat data ze zařízení. Tento cíl jsem splnil přesně jak bylo zamýšleno. Dalším zásadním cílem byla možnost jednoduchého ovládání hudby a zvuku při ovládání robota tento cíl se mi povedlo splnit bohužel ne dle původních představ a to z důvodu autorských práv na zamýšlené hudební stopy. Tento problém jsem vyřešil použitím stop pod GNU GPL licencí.

10. Použitá literatura

Kořínek, Lukáš. Programovací jazyk Swift a účelnost jeho zařazení do výuky [online]. 4. 5. 2016 [cit. 2018-02-25]. Dostupné z: https://vskp.vse.cz/50629_programovaci_jazyk_swift_a%C2%A0ucelnost_jeho_zarazeni_do_vyuky

SPI interface Tutorial. [online]. [cit. 2018-02-25]. Dostupné z: <http://www.best-microcontroller-projects.com/spi-interface.html>

In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-02-25]. Dostupné z: http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

SPI - A simple implementation. Fpga4fun.com [online]. [cit. 2018-02-25]. Dostupné z: <http://www.fpga4fun.com/SPI2.html>

In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-02-25]. Dostupné z: <http://en.wikipedia.org/wiki/I2C>

I²C. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-02-25]. Dostupné z: https://en.wikipedia.org/wiki/Clock_signal

Externí sériové sběrnice SPI a I²C. ROOT.CZ [online]. 2008, 30. 12. 2008 [cit. 2018-02-25]. Dostupné z: <https://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/>

Icons8 [online]. [cit. 2018-02-25]. Dostupné z: <https://icons8.com/icon/new-icons/ios>

GitHub [online]. [cit. 2018-02-25]. Dostupné z: <https://github.com/>

Freesound: Sounds downloaded by Legiani [online]. [cit. 2018-02-25]. Dostupné z: https://freesound.org/people/Legiani/downloaded_sounds/

The Swift Programming Language (Swift 4.0.3) [online]. Apple, 2014 [cit. 2018-02-25]. Dostupné z: <https://itunes.apple.com/us/book/the-swift-programming-language-swift-4-0-3/id881256329?mt=11>

The Swift Programming Language (Swift 4.1 beta) [online]. Apple, 2015 [cit. 2018-02-25]. Dostupné z: <https://itunes.apple.com/us/book/the-swift-programming-language-swift-4-1-beta/id1002622538?mt=11>

About Core Bluetooth. Guides and Sample Code [online]. 2013 [cit. 2018-02-25]. Dostupné z: https://developer.apple.com/library/content/documentation/NetworkingInternetWeb/Conceptual/CoreBluetooth_concepts/AboutCoreBluetooth/Introduction.html#//apple_ref/doc/uid/TP40013257

hoiberg. HM10 Bluetooth Serial iOS. GitHub [online]. 2015 [cit. 2018-02-25]. Dostupné z: <https://github.com/hoiberg/HM10-BluetoothSerial-iOS>

Xcode: Xcode. Guides and Sample Code [online]. [cit. 2018-02-25]. Dostupné z: <https://developer.apple.com/library/content/navigation/#section=Topics&topic=Xcode>

Stackoverflow: Learn, Share, Build [online]. [cit. 2018-02-25]. Dostupné z: <https://stackoverflow.com>

The Untold History of Arduino: Index [online]. [cit. 2018-02-25]. Dostupné z: <https://arduinohistory.github.io>

BHATT, Nitin. Oodles Technologies: Open iTunes Music Library to Play Songs in Swift. Oodles Technologies: Open iTunes Music Library to Play Songs in Swift [online]. [cit. 2018-02-25]. Dostupné z: <http://www.oodlestechnologies.com/blogs/Open-iTunes-Music-Library-to-Play-Songs-in-Swift>

AFANEH, Mohammad. Bluetooth GATT: How to Design Custom Services & Characteristics [MIDI device use case]. NovelBits [online]. 2017, 2017 [cit. 2018-02-25]. Dostupné z: <http://www.novelbits.io/bluetooth-gatt-services-characteristics/>

11.Seznam zkratek

BLE - Bluetooth

MSE - 6 - Mause droid tipu 6

PAN - Personal area network

GNU GPL - Licence pro volné použití

PC - počítač

IDE - vyvojové prostředí (editor)

HW - hardware

SW - software

OS - operační systém

App - aplikace

12.Seznam obrázku

