

**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

**FAKULTA
STROJNÍ**



**BAKALÁŘSKÁ
PRÁCE**

2019

**NURLAN
TURGANBAY**

České vysoké učení technické v Praze
Fakulta strojní

Ústav přístrojové a řídicí techniky
Obor: Informační a automatizační technika

**Mobilní aplikace pro řízení ruky robota s taktilním
čidlem**

**Mobile application for wireless control of robot's
hand with a tactile sensor**

BAKALÁŘSKÁ PRÁCE

Vypracoval: Nurlan Turganbay
Vedoucí práce: Ing. Zdeněk Novák
Rok vypracování: 2019

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Turganbay** Jméno: **Nurlan** Osobní číslo: **438333**
Fakulta/ústav: **Fakulta strojní**
Zadávající katedra/ústav: **Ústav přístrojové a řídicí techniky**
Studijní program: **Strojirenství**
Studijní obor: **Informační a automatizační technika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Mobilní aplikace pro řízení ruky robota s taktilním čidlem

Název bakalářské práce anglicky:

Mobile application for wireless control of robot's hand with a tactile sensor

Pokyny pro vypracování:

- 1) Zpracujte rešerši na téma 'Taktilní čidla a využití těchto senzorů v průmyslu a zdravotnictví, použitelnost ve spojení s mobilní aplikací'.
- 2) Navrhněte způsob umístění taktilního čidla na ruku robota. Vytvořte vlastní mobilní aplikaci pro bezdrátové řízení této ruky, která bude umět i zobrazovat informaci o síle úchopu jednoduchého předmětu v reálném čase.
- 3) Aplikaci otestujte a navrhněte další možná rozšíření a použití.

Seznam doporučené literatury:

- [1] WOLBER, David. App Inventor: create your own Android apps. Sebastopol, Calif.: O'Reilly, 2011, ISBN 14-493-9748-4.
- [2] DUFFY, Thomas J. Programming with mobile applications: Android, iOS, and Windows Phone 7. Boston, MA: Course Technology/Cengage Learning, 2013. ISBN 11-336-2813-3
- [3] HORTON, John. Android programming for beginners, Packt Publishing, 2015, ISBN-13: 978-1785883262
- [4] FRADEN, Jacob. Handbook of modern sensors: physics, designs, and applications. 4th ed. New York: Springer, c2010, ISBN 978-1-4419-6465-6.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Zdeněk Novák, U12110.1

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


Datum zadání bakalářské práce: **26.04.2019**

Termín odevzdání bakalářské práce: **12.06.2019**

Platnost zadání bakalářské práce:



Ing. Zdeněk Novák
podpis vedoucí(ho) práce



podpis vedoucí(ho) ústředí/katedry



prof. Ing. Michael Valášek, DrSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

26-04-2019

Datum převzetí zadání



Podpis studenta

Prohlášení:

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně s tím, že její výsledky mohou být dále použity podle uvážení vedoucího bakalářské práce jako jejího spoluautora. Souhlasím také s případnou publikací výsledků bakalářské práce nebo její podstatné části, pokud budu uveden jako její spoluautor.

V Praze dne:

Podpis:

Poděkování

Chtěl bych poděkovat panu Ing. Zdeňku Novákovi za jeho odborné konzultace a rady při vedení této bakalářské práce. Děkuji také mé rodině za finanční a psychickou podporu během celého studia.

Název: Mobilní aplikace pro řízení ruky robota s taktilním čidlem

Abstrakt

Tato bakalářská práce je zaměřena na řízení robotické ruky s taktilním čidlem pomocí vlastní aplikace. Práce se skládá ze tří částí. Nejdříve je rešerše na toto téma. Jsou popsány typy taktilních čidel a Arduina, způsoby bezdrátové řízení dat a jazyky programování. Druhá část je věnována umístění čidla, zapojení všech součástí a vytvoření vlastní mobilní aplikace. Ve třetí části jsou popsány výsledky otestování a případně návrhy na zlepšení.

Klíčova slova: Taktilní čidla, FSR400, Arduino, Bluetooth, Android Studio, Vývojové prostředí, řízení, aplikace.

Title: Mobile application for wireless control of robot's hand with a tactile sensor

Abstract

This bachelor thesis is focused on control of robotic hand with tactile sensor using own application. The thesis consists of three parts. First, a research on this subject is made. Follows a description of types of tactile sensors and Arduino, methods of wireless data management and programming languages. The second part is devoted to the location of the sensor, the integration of all components and the creation of its own mobile application. The third part describes the results of testing and possibly suggestions for improvement.

Keywords: Tactile sensors, FSR400, Arduino, Bluetooth, Android Studio, IDE, control, application.

Obsah

Úvod.....	7
1. Rešerše	8
1.1 Taktilní čidla, definice pojmu	8
1.2 Typy taktilních čidel	8
1.3 Taktilní čidla FSR400.....	9
1.4 Využití čidel v průmyslu a ve zdravotnictví	11
1.4.1 Taktilní čidla v průmyslu.....	11
1.4.2 Taktilní čidla v medicíně	12
2. Vývojové prostředí	13
2.1 Arduino	13
2.1.1 Mikrokontrolér.....	13
2.1.2 Digitální a analogové piny vstupu a výstupu	14
2.2 Vývojové prostředí pro vytvoření mobilní aplikace	14
2.2.1 Jazyky programování pro vytvoření aplikace	14
2.2.2 Android Studio	16
3. Moduly pro bezdrátové řízení.....	18
3.1 Bluetooth.....	18
3.2 Wi-Fi.....	19
4. Praktická část.....	20
4.1 Zprovoznění ruky robota.....	20
4.2 Umístění čidla na dlani.....	20
4.3 Převedení odporu na sílu	21
4.4 Testování Bluetooth modulu	22
5. Zapojení a programování.....	24
5.1 Popis programu v Arduino IDE.....	24
5.2 Vytvoření mobilní aplikace	28
6. Testování funkčnosti aplikace a kódu	33
7. Návrhy pro zlepšení a použití.....	35
8. Závěr.....	36
Literatura	37

Úvod

V dnešní době rychle roste potřeba elektrických zařízení. Všechny oblasti průmyslu se postupně automatizují. Poměr mezi roboty a dělníky neustále narůstá. Je to způsobeno tím, že použití robotů je mnohem výhodnější.

Jedním z nejpoužívanějších typů senzorů v průmyslu jsou taktilní čidla. Jednoduše řečeno se jedná o senzory, které převádějí fyzickou akci na elektrický signál. Použití tenzometrických přístrojů neustále roste. V současné době jsou tenzometry široce používány v automobilovém průmyslu, ve výrobě letadel, v medicíně, v biomechanickém inženýrství a v robotice.

Účelem této práce je ovládání ruky robota bezdrátovým řízením pomocí mnou vytvořené aplikace. Jedním z hlavních podcílů je způsob umístění taktilního čidla tak, aby ruka mohla uchopit předmět, a aby se přitom předmět nezničil.

První část práce se zabývá rešerší informací o taktilních čidlech, bezdrátových modulech a hledání základní informací pro vytvoření aplikace.

Ve druhé části se budu zabývat instalací čidla na ruku robota, a prvními kroky komunikace s Bluetooth modulem. Hlavním cílem této části je napsat program ve vývojových prostředích.

Dalším cílem je otestovat programy pro řízení ruky a popřípadě návrhy na jejich zlepšení.

1. Rešerše

1.1 Taktilní čidla, definice pojmu

Taktilní čidla jsou zařízení, které jsou schopny přeměnit mechanickou deformaci na elektrický signál. Díky těmto elektrickým signálům vzniká možnost získat informaci o dotyku. Jinými slovy pomocí taktilních čidel můžeme pozorovat interakci člověka se zařízením. Nyní jsou tato čidla používána v automatizovaných systémech všech sfér průmyslu. Výhoda těchto senzorů je v tom, že jsou malé a lehké.

Taktilní čidla se skládají z citlivých tenzometrů, která jsou vyrobená z tenzomateriálu. Většinou je to fólie nebo hliníkový drát. Tyto citlivé tenzometry, stejně jako normální kuchyňská váha reagují na deformace povrchu [1].

1.2 Typy taktilních čidel

Existují různé druhy taktilních čidel. Podle parametrů a principu lze taktilní čidla rozdělit na tyto skupiny – Taktilní čidla:

- S elastomery;
- s piezoelektrickými materiály;
- s optickými vlákny;
- kapacitní.

Taktilní čidla s elastomery

Čidla s elastomery se dělí na: s vodivým elastomerem, čidla typu FSR, s odporovou vrstvou a detekční fólie Pressurex. Princip čidla s vodivým elastomerem spočívá v tom, že s působící silou vodivý elastomer mění svůj elektrický odpor. Poté se elektrický odpor přemění na elektrický signál. Taktilní čidlo s vodivým elastomerem typu CS 57-7 RSC umožňuje měřit i velikost tlaku [6].

Čidla s odporovou vrstvou reagují na normálové a tečné síly. Ve skutečnosti se jedná o dvě propojené vrstevové snímače. První vrstva, která vytváří dynamický senzor je tvořena klky. Tento senzor je citlivý na dynamické změny [6]. Dynamické změny jsou způsobeny působením tečné složky síly, na kterou spodní vrstva nereaguje.

Taktilní čidla s piezoelektrickými materiály

Prvkem citlivosti v tomto senzoru je piezoelektrický prvek. Jedná se o látku, která během deformace vytváří elektrický signál. Tato vlastnost se nazývá přímý piezoelektrický efekt. Jako piezoelektrické materiály jsou používány: krystaly titaničitanu barnatého, turmalínu, křemene. Jsou chemicky odolné, mají vysokou pevnost a jejich vlastnosti jsou málo závislé na okolní teplotě. V měřené oblasti je piezoelektrický prvek, který tvoří proud přímo závislý na hodnotě tlaku. Signál ve snímači piezoelektrických materiálů je tvořen pouze během deformace. Při konstantním tlaku nedochází k deformaci, proto je senzor vhodný pouze pro měření média s rychle se měnícím tlakem. Pokud se tlak nezmění, pak nedojde k žádné deformaci a piezoelektrický signál nebude generován [4].

Taktilní čidla s optickými vlákny

Snímače tlaku s optickými vlákny jsou nejpřesnější a jejich provoz není závislý na výkyvech teploty. Citlivý prvek je optický vlnod. Měřený tlak v těchto zařízeních je obvykle posuzován změnou amplitudy a polarizací světla procházejícího citlivým prvkem. Optické vláknové senzory nejsou citlivé na vnější elektromagnetická pole a umožňují přenášet informaci mezi objekty s rozdílnými elektrickými potenciály v širokém pásmu kmitočtů a na velké vzdálenosti [6].

Kapacitní taktilní čidla

Jedná se o parametrický typ snímače, kterým je kondenzátor. Má jeden z nejjednodušších konstrukcí. Skládá se ze dvou ploch elektrod a mezery mezi nimi. Čím větší je plocha elektrod, tím větší je kapacita. Čím větší je mezera mezi plochami, tím menší je kapacita. Měří se deformace pružného členu mezi dvěma plochami při působení síly. To je v podstatě tento typ snímače „*kondenzátor s proměnnou mezerou*“. Kapacitní snímače mohou detekovat velmi malé změny tlaku.

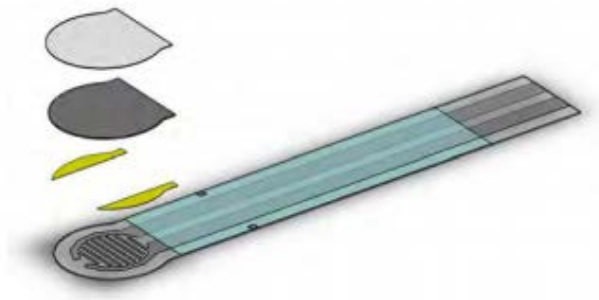
Tato vlastnost se používá pro návrh kapacitních senzorů. Aby bylo možné měření pohodlně používat, kapacita se přemění na proud.

1.3 Taktilní čidla FSR400

Tato čidla jsou používána v elektrických zařízeních s dotykem ruky, v automatizované elektronice a v medicíně. Ale častěji v robotice, pro určení síly stlačení. Například, aby robot mohl mírně uchopit nějaký předmět.

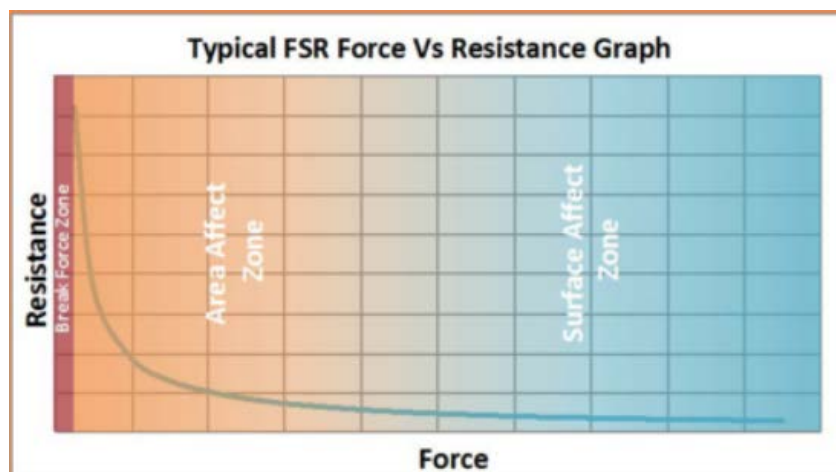
FSR (Force Sensing Resistor) jsou čidla, které umožňují stanovit sílu stlačení a váhu. Pro stanovení deformací jsou využívány následující typy kontaktů: fóliové, povlakové a drátkové. Povlaková čidla se skládá ze dvou povlakových vrstev (Obrázek č.1). Jsou odděleny speciálním těsněním (spacer) [2].

Tímto způsobem čím víc tlačíme, tím líp se stává styk mezi dvěma vrstvy. Princip je v tom, že při deformaci čidlo mění svůj vlastní odpor, a přemění odpor na elektrický signál. V ustáleném stavu je odpor čidla příliš velký, až 10 MOhm. Při tlačení na senzor odpor klesá až do 2-10 kOhmu. Tyto hodnoty jsou různé v závislosti na typu čidla rodiny FSR.



Obrázek č.1 — Taktilní čidlo FSR 400 [2].

Změna odporu je neproporcionální k síle tlačení. Proto při práci s tímto čidlem lze vidět 3 oblasti citlivosti. První oblast se jmenuje Break force. V této oblasti je citlivost čidla moc vysoká, při slabém tlačení odpor klesá přibližně do 100 KOhm. Pokud se na čidlo přidá síla, pak lze dosáhnout druhé oblasti, která se nazývá Area Effect. Citlivost v této oblasti je nízká, skoky odporu se ztratí a závislost odporu na síle se stává víceméně plynulou. Obvykle odpor kmitá v intervalu od 25 KOhm do 100 KOhm. Třetí oblast citlivosti se nazývá Surface Effect. Oblast citlivosti, ve které je působící síla pro čidlo moc vysoká, zvětšení síly je nevýznamné na změny odporu.



Obrázek č. 2 — Závislost odporu na síle [2].

Tento vnitřní odpor je nelineární na působenou sílu v první a třetí oblasti. Jen ve druhé oblasti dá se víceméně použít charakteristiky senzoru. Jelikož se jedná o čidlo nelineární, nemůžeme měřit sílu přesně. Nelze to použít pro určení hmotnosti. Ale ve sférách, kde není nutné měřit přesnost až do gramu, je toto čidlo možné použít.

1.4 Využití čidel v průmyslu a ve zdravotnictví

Tato čidla se používají v různých odvětvích průmyslu a zdravotnictví. Například piezoelektrické senzory se používají pro záznam lidského pulsu, protože jsou vhodné pro měření dynamických sil, resp. vibrací. Současně jsou dnes kapacitní snímače nejběžnější polovodičová zařízení pro získávání otisků prstů [4].

1.4.1 Taktilní čidla v průmyslu

Robotika představuje hojné využívání taktilních informací v průmyslu. Na začátku byla taktilní čidla používány jen pro indikace úchopu předmětu rukou, pak pro řízení úchopu a měření síly a polohy [6].

Taktilní čidla přímo realizují bionickou funkci dotyku a jsou určeny pro geometrické rozpoznání vnějšího prostředí při kontaktu [6]. Škála možných aplikací dotykových senzorů je velmi široká – od úkolů zajištění bezpečnosti při práci s příslušnými objekty, až po rozpoznávání vzorů částí složitého tvaru. Pokud jde o průmyslovou výrobu, tato zařízení se používají například při sledování trajektorie při obloukovém svařování; odstranění částí z kontejneru a adaptivní zachycení křehkých předmětů [23].

V robotice byly taktilní čidla první nástroje cítění – již v roce 1961 H. Ernst jimi vybavil „mechanickou ruku“ MH-1 [23]. Od té doby jsou taktilní čidla důležitým nástrojem pro přizpůsobení robotů. V současné době mají taktilní čidla především speciální aplikace – při úlohách dálkového ovládání manipulátorů, mobilních robotických prostředků, zařízení pro přenášení a skladování [23].

Taktilní čidla systému robota řeší následující hlavní úkoly:

- Detekce kontaktu nástroje s objektem;
- určení souřadnic a oblasti kontaktního místa;
- měření uchopovací síly;
- detekce sklouznutí a měření posunutí;
- rozpoznání objektů jejich taktilním obrazem.

1.4.2 Taktilní čidla v medicíně

Ruční protézy

Taktilní čidla hrají velmi důležitou roli ve vývoji moderní medicíny. Inovace přišly do té míry, že speciální implantáty jsou implantovány přímo do nervového systému, pro částečné vrácení přirozené schopnosti člověka. Inženýři na Stanfordské univerzitě vytvořili plastovou „kůži“, která používá tenzometry k určení síly dotyku (senzory jsou namontovány na špičkách pěti prstů) a generují elektrický signál, který přenáší tuto informaci do mozku [22]. Vynález zahrnuje síť senzorů, které vnímají a vysílají informace do mozku o síle dotyku, teplotě a bolesti. Tato inovace pomáhá pacientovi získat protézu, která bude působit téměř jako "nativní" ruka [22].

Zdravotní lůžka

Zdravotní lůžka jsou určena pro péči a léčbu pacientů pod dohledem lékaře. Tenzometry zabudované do lůžka pomáhají kontrolovat hubnutí a přibývání na váze, kontrolují distribuci tělesné hmotnosti, řídí citlivost spánku a mnoho dalších ukazatelů.

Taktilní čidla ve stomatologii

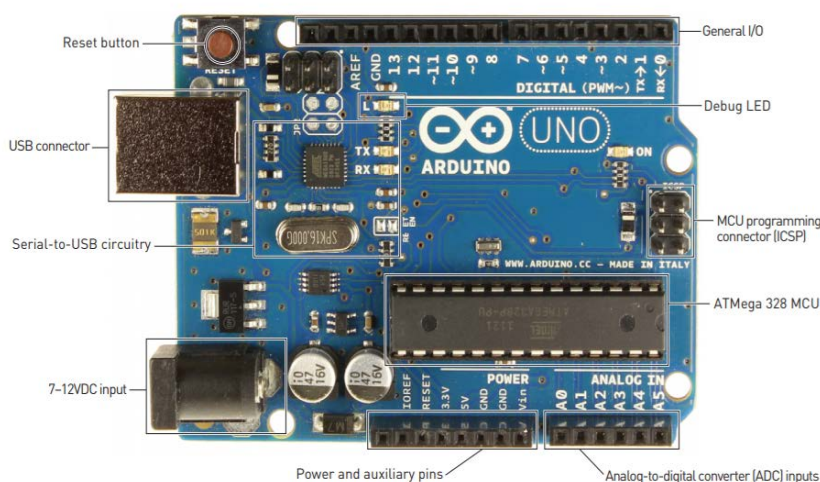
Univerzity se zaměřením na stomatologický výzkum používají tenzometry k měření zatížení žvýkacího povrchu zubů, stejně jako ke zkoumání rozložení úsilí mezi protetickým lůžkem a základem protéz různých vzorů [22]. Výzkum prováděné pomocí tenzometrů umožňují posoudit stav protetického lůžka a rozložení zátěže na jednotlivé body, což je důležitým ukazatelem kvality zubní protetiky. Použití kmene umožňuje získat cenné informace během přípravy, protože umožňuje předvídat nejen kvalitu zpracování, ale i dlouhodobé výsledky zubní protetiky [22].

2. Vývojové prostředí

2.1 Arduino

Arduino – je otevřená elektronická platforma, založená na snadno použitelném hardwaru a softwaru [9]. Pomocí Arduino můžeme realizovat širokou škálu aplikací. Může to být automatické řízení ventilátoru, nebo automatické ukládání nových publikací na webové stránky. Arduino je platforma pro vývoj zařízení na základě mikrokontroléru v jednoduchém jazyce programování v integrovaném prostředí Arduino IDE. K Arduino můžeme připojit snímače teploty, snímače světla, snímače pohybu, reproduktory, dodatečné moduly atd. Těžko jde předvést, co všechno je schopná platforma Arduino, protože možnosti jsou omezeny pouze Vaší kreativitou [9].

Platforma Arduino se skládá z desky, mikrokontroléru, operační paměti, z digitálních a analogových pin vstupů a výstupů. Takže v některých typech desek je USB-port a konektor pro napájení od vnějšího zdroje.



Obrázek č. 3 — Arduino Uno [9].

2.1.1 Mikrokontrolér

Mikrokontrolér – je malá součástka s mikro schématem. Na monolitickém integrovaném obvodu mikrokontroléru je instalován mikropočítač. To znamená, že uvnitř toho mikrokontroléru je obsažen procesor, operační paměť a periferní zařízení, které mezi sebou budou spolupracovat pomocí speciálního softwaru, který bude uložen v mikrokontroléru. Mikrokontroléry firmy Atmel jsou srdcem všech platform Arduino. Většinou se používají typy ATmega328, ATmega168, ATmega2560, ATmega16U2 a ATmega32U4.

2.1.2 Digitální a analogové piny vstupu a výstupu

Platforma Arduino má digitální a analogové vstupy a výstupy, ke kterým můžeme připojit vnější periférie.

Třeba termočláanky, nebo snímače dotyku. Všechny piny můžou sloužit jako digitální vstupy a výstupy. Některé digitální piny lze použít pro *PŠM* (pulzně šířková modulace). Pulzně šířková modulace je diskretní modulace pro přenos analogového signálu pomocí dvouhodnotového signálu.

2.2 Vývojové prostředí pro vytvoření mobilní aplikace

Popularita mobilních aplikací v dnešním světě prudce roste. Dříve častěji používané aplikace v počítačích byli už dávno vytvořené i pro mobilní telefony. Což vypovídá o tom, že se mobilní telefony stávají více prioritními než počítače. To samé platí i v průmyslu, lidé ladně přecházejí na řízení pomocí mobilních zařízení než z velkých panelů. Aby bylo možné ovládat stroje zařízeními, je nutné vytvořit aplikace na platformě. Tvorba aplikací je dále probírána níže.

2.2.1 Jazyky programování pro vytvoření aplikace

Jazyky programování pro vytvoření aplikací na platformě Android jsou:

- Java;
- Kotlin;
- C/C++;
- C#;
- BASIC;
- Python;
- Lua;
- PhoneGap.



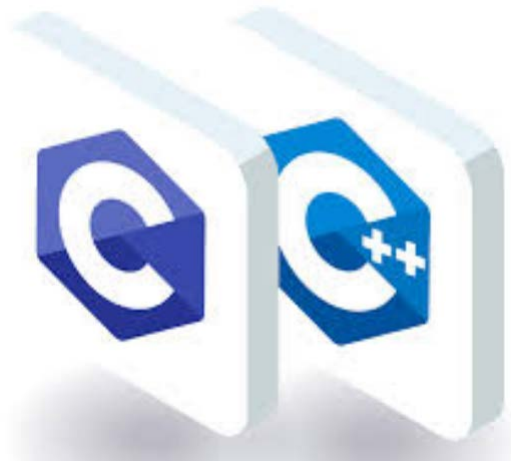
Obrázek č. 4 — Firemní značka Java [20].

Jazyk programování Java je oficiálním jazykem Android. Skoro všechna výuková dokumentace je na něm založená. Je jedním z nepopulárnějších jazyků programování, a většina výchozích materiálů na GitHub jsou na Java. Proto je pro Android vývojáře jedním ze základních jazyků pro výuku. Nejlehčí způsob vytvoření aplikace v Java je stáhnout integrované vývojové prostředí (IDE) Android Studio [17].



Obrázek č.5 — Programovací jazyk Kotlin [20].

Programovací jazyk Kotlin byl představen v roce 2017, a je počítán jako druhý oficiální jazyk programování pro Android. Je nutné umět jazyk Java, aby bylo možné pochopit základní princip jazyka Kotlin, jeho strukturu a možnosti.



Obrázek č.6 — Jazyky C a C++ [21].

Jazyky C a C++ jsou nižší programovací jazyky. Můžou být použity v Android Studio přes Android NDK. Pro začátečníky je vytvoření aplikací s jazyky C a C++ je mnohem těžší. Kód napsaný v C nebo C++ nelze přímo spustit v JVM (Java Virtual Machine), ale lze spustit na zařízení. Je vhodný pro napsání knihovny.

2.2.2 Android Studio

Android Studio je oficiální vývojové prostředí založené společností Google. Většina aplikací, které jsou používány každý den, byli vytvořeny v tomto prostředí. Poprvé bylo Android Studio objeveno v roce 2013. Předtím byla většina aplikací rozvíjena v Eclipse. Android Studio umožňuje vývoj aplikací jednoduše.



Obrázek č.7 — Logotyp Android Studio [17].

Android Studio je dodáván s Android SDK (Software Development Kit). Je to sada instrumentů pro vývoj Android aplikací, která umožňuje bez složitostí začít pracovat. Jedním z důvodů, proč používat Android Studio je ten, že v prostředí je přímo nainstalován emulátor telefonu. V emulátoru lze hned otestovat napsaný kód, což urychluje práci. Android Studio ukazuje různé rady při psaní kódu a vyhlašuje případné chyby.

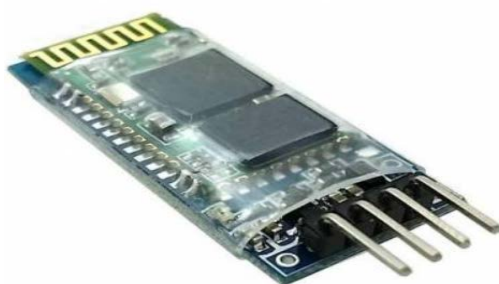
3. Moduly pro bezdrátové řízení

Bezdrátové řízení je jedním ze způsobu komunikace s vedeným zařízením. V dnešní době jsou bezdrátová zařízení velice populární. Je to svázáno s tím, že ovládání zařízení z dálky je uživatelsky příjemné. Vzniká mnoho nových projektů, jako je například chytrý dům, bezdrátový vypínač světla atd. Bezdrátové řízení jsou většinou realizována metodami infračervené, ultrazvukové, Bluetooth (dále jen BT), GPS a Wi-Fi přenosu dat. Podkapitoly níže budou věnovány pouze BT a Wi-Fi.

3.1 Bluetooth

Pokud přeložíme slovo BT z angličtiny, zjistíme že znamená „modrý zub“. Tento pojem nebyl vybrán náhodou. Faktem je, že v roce 900 v Dánsku žil král Harald Bluetooth. Tento král využil své diplomatické schopnosti ke sdružení dvou států – Norsko a Dánsko. Proto společnost Ericsson, na jeho počest, pojmenovala jejich novou technologii jeho jménem Bluetooth [24].

V této technologii probíhá přenos dat pomocí bezdrátových kanálů. BT umožňuje vytvářet bezdrátové komunikace mezi zařízeními, díky kterým můžeme přenášet jakoukoliv informaci. Kromě toho pomocí BT můžeme řídit zařízení jako sluchátka, tiskárny, skenery atp. Princip BT spočívá v tom, že přemění digitální signál na analogový a přenáší jej přes bezdrátové kanály na jiné zařízení. Druhé zařízení přemění signál zase na digitální. Přenos dat probíhá ve kmitočtových pásmech *ISM*, které nepotřebují licenci.



Obrázek č. 8 — BT modul HC-06 [13].

3.2 Wi-Fi

Technologie bezdrátové sítě funguje na základě standartu IEEE 802.11. Tento pojem pochází z anglického slovního spojení Wireless Fidelity, což znamená bezdrátový přenos. Stejně jako BT používá tato technologie kmitočtová pásma bez licencí. Nyní se Wi-Fi používá všude, v posledních 10 letech je použita i pro řízení mikroprocesorů [25].

Existuje spousta typů Wi-Fi modulů pro mikroprocesory. Čínská společnost Espressif v roce 2014 objevila modul ESP na čipu ESP8266. Zařízení se stalo ihned populární mezi fanoušky elektrických zařízení, kvůli své ceně a možnostem použití. V dnešní době vyrábí společnost různé typy modulů ESP. Jedním z nich je ESP-01. Čip ESP8266 představuje mikroprocesor s Wi-Fi vysílačem, který může fungovat i bez řídicích mikroprocesorů (např. Arduino). Pomocí modulu ESP-01 lze přenášet data o teplotě, vlhkosti nebo řídit zařízení. Pro práci s modulem ESP-01 je nutné externí napájecí zdroj, který předává napětí na 3,3 V a proud minimálně na 250 mA.

4. Praktická část

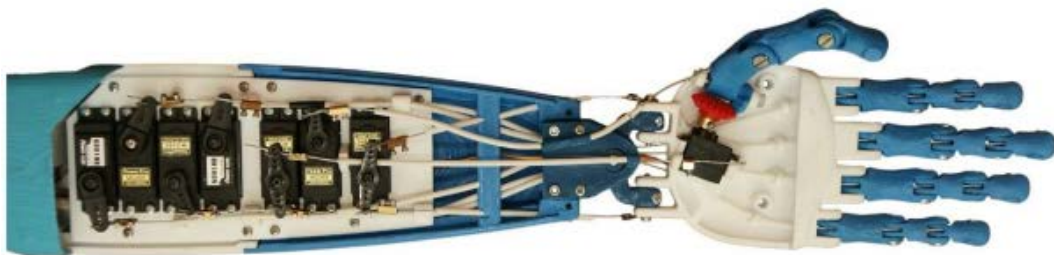
Praktická část této bakalářské práce je zaměřena na umístění taktilního čidla a na vytvoření vlastní mobilní aplikace pro bezdrátové řízení ruky.

Pro realizaci praktické části jsem použil následující komponenty:

- Arduino Mega2560;
- taktilní čidlo FSR400;
- Servomotory;
- Bluetooth module HC-05;
- Propojovací vodiče.

4.1 Zprovoznění ruky robota

Tato ruka byla udělaná v rámci bakalářské práce jiného studenta. Na první pohled je ruka v dobrém stavu, ale ve vnitřních součástech ruky bylo pár drátů roztrhaných a některé servomotory spálené. Odpojil jsem všechny součásti ty ruky a vyměnil nefungující serva za nové. Vyndal jsem šrouby na špičce prsténíčku a ukazováčku. Poté jsem vyndal roztrhané dráty u dvou prstů, a vyměnil je za nové.



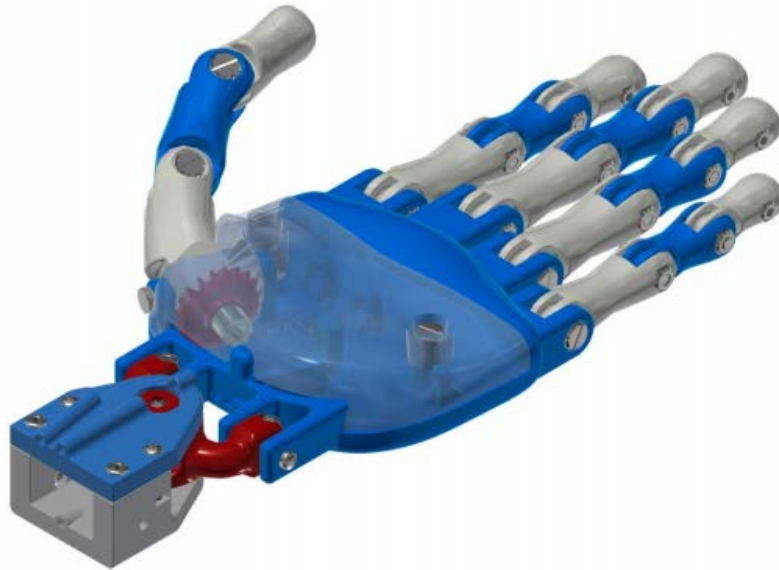
Obrázek č. 9 — Humanoidní ruka robota [16].

4.2 Umístění čidla na dlani

Po opravě robotické ruky jsem přemýšlel, kam nejlépe umístit čidlo, aby měřilo hodnoty co nejpřesněji. V rámci bakalářské práce používám taktilní čidlo FSR400. Toto čidlo nejde umístit na špičky prstů, protože čidlo má příliš dlouhé vývody. Po několika cyklech zatnutí a uvolnění pěsti by čidlo pravděpodobně ztratilo svou kvalitu. Ideálně by bylo, kdybych použil FSR400

Short. Krátké tvary čidla FSR400 jsou vhodné pro umístění na špičky prstů a bylo by možné měřit sílu každého prstu zvlášť.

Kvůli tomu, že nemohu umístit čidlo na špičky, musím ho umístit jinak. Abych neztratil přesnost měření musím čidlo umístit doprostřed. Všechny palce budou řízeny pomocí jednoho čidla. Vytvořená dlaň není rovinná, proto aby ruka mohla uchopit i kulaté věci. Z tohoto důvodu jsem se rozhodl, že čidlo umístím uprostřed a blízko k prstům.



Obrázek č. 10 — 3D model dlaně [16].

4.3 Převedení odporu na sílu

Taktilní čidlo FSR 400 má být připojeno pouze na analogový výstup. Mezi analogovým pinem a zemí zapojím odpor na 100 kOhm. Když tlačím na čidlo, na analogovém výstupu dostávám pouze počet bitů. Poté počet bitů převedu na výstupní napětí. Po úpravě vzorce z technické dokumentace dostanu vnitřní odpor:

$$R_{FSR} = \frac{R_M \cdot (V_+ - V_{OUT})}{V_{OUT}}$$

R_M je odpor 100 kOhm

V_+ je napájení na 5000 mV

V_{OUT} je výstupní napětí (mV)

R_{FSR} je vnitřní odpor senzoru

Poté pomocí převrácené hodnoty odporu, tzv. konduktance, zjistím sílu stlačení. To všechno provedu v programovací prostředí Arduino IDE. Pomocí příkazu `Serial.begin` zjistím aktuální sílu v Port Monitoru.

```
#define FSR_PIN A0          // pin na které zapojeno čidlo
int val;                  // počet bitů
int Rm;                   // odpor na 100K
int U_out;                // výstupní napětí
unsigned long Rfsr;       // vnitřní odpor senzoru
unsigned long Conductance; // převrácená hodnota odporu
float Force;              // hledaná síla

void setup() {
  Serial.begin(9600);      //Senzor bude posílat sem aktuální hodnoty
}

void loop() {
  val = analogRead(FSR_PIN); // čtu hodnotu z pinů A0
  U_out = map(val, 0, 1023, 0, 5000); //5000mV = 5V

  Rfsr = 5000 - U_out;
  Rfsr *= 100000;
  Rfsr /= U_out;          // vzorec pro získání vnitřního odporu
  Serial.print("FSR odpor v ohmech = ");
  Serial.println(Rfsr);

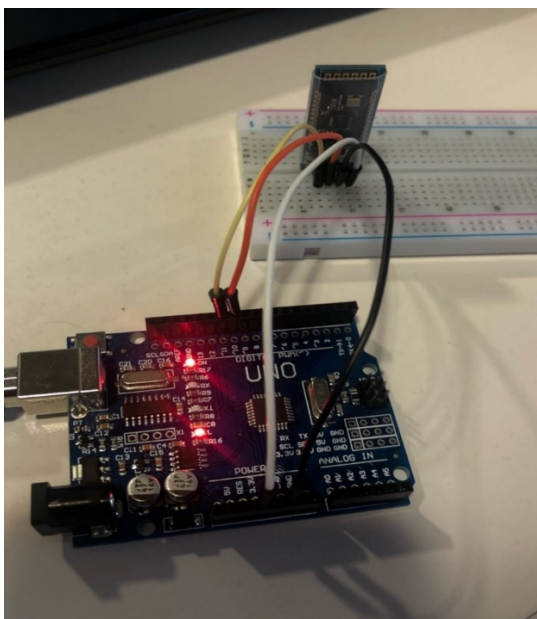
  Conductance = 1000000; // budeme měřit v  $\mu\Omega$ 
  Conductance /= Rfsr;

  Force = Conductance / 80;
  Serial.print("Síla v N = ");
  Serial.println(Force);
}
```

Obrázek č. 11 — Kód pro určení aktuální síly

4.4 Testování Bluetooth modulu

Před zapojením a programováním je nutné ověřit, zda Bluetooth modul funguje správně. Jednoduchým způsobem je použití Serial monitor v prostředí Arduino IDE. Stačí zkompileovat jednoduchý kód a pomocí AT příkazů pro Bluetooth zjistit funkčnost modulu. V rámci bakalářské práce jsem použil Bluetooth HC-05. Zapojení se provádí následovně:



Obrázek č. 12 — Zapojení pro ověření funkčnosti Bluetooth.

Na obrázku je černý drát definován jako uzemnění (GND), bílý drát je napájení (Vcc=5 V). Oranžové a žluté dráty jsou pro vysílání a příjem dat (TX, RX). Oranžový drát zapojen k pinu 10 který jsem v Arduino definoval jako RX, žlutý pin naopak. Důležitou věcí je, že RX má být zapojen do TX, TX naopak od RX, jinak Serial monitor nic neukáže.

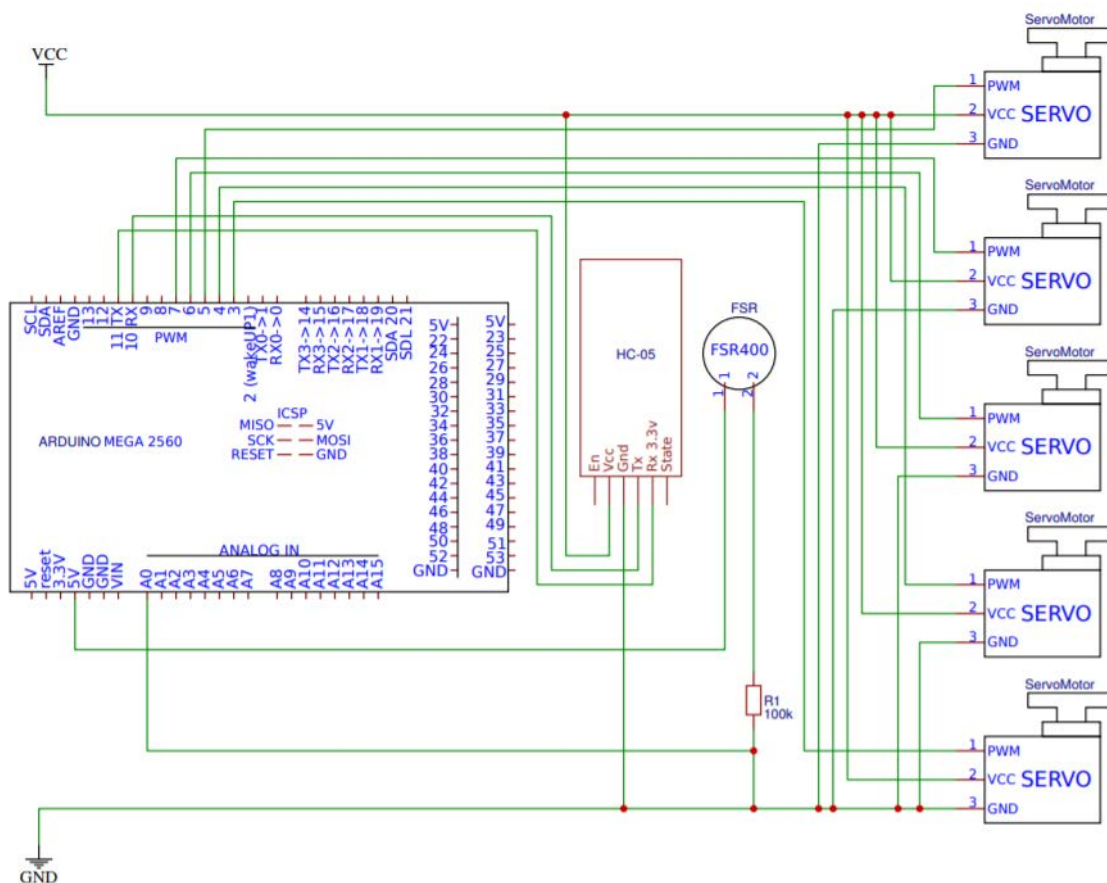
Modul HC-05 má ještě 2 piny. Jeden z nich je *STATE* pin, který indikuje stav modulu. Druhý je pin *KEY* nebo *EN*, který je důležitý pro vstup do režie AT-příkazu. Pro převedení modulu do režie AT-příkazu je nutné zapojit KEY pin, poté připojit Arduino do počítače a odpojit KEY pin. Po správném pořadí chodu, by mělo led světlo na modulu blikat jinak. Po zapojení a zkompilování kódu stačí otevřít Seriál monitor a poslat příkaz „AT“, pokud je vše zapojeno správně, tak je vidět na monitoru odpověď „OK“. Pomocí AT-příkazů můžeme nastavit základní konfigurace, například nastavit heslo a název. Po ověření funkčnosti modulu jsem potřeboval zjistit, zda se k němu mohu připojit. Proto jsem se z telefonu zkusil připojit, a výsledek má vypadat takto:

```
COM3
AT+NAME=RoboHand
OK
+VERSION+=BOLUTEK Firmware V2.2, Bluetooth V2.1
+CONNECTING<<C0:11:73:70:78:34
CONNECTED
```

Obrázek č. 13 — Komunikace s monitorem

5. Zapojení a programování

Táto kapitola bude věnovaná popisu kódu v prostředí Arduino IDE a tvoření aplikace v Android Studio. Po otestování Bluetooth modulu a taktilního čidla je nutné připojit do Arduina. Na obrázku č. 14 znázorněno správné zapojení.



Obrázek č. 14 — Schéma zapojení.

5.1 Popis programu v Arduino IDE

Po zapojení všech motorů, BT modulu a taktilního čidla zbylo pouze napsat hlavní program, který bude řídit ruku. K tomu jsem použil Arduino Mega2560, protože je výkonnější a má více pinů než ostatní Arduino. Na začátku jsem nadefinoval taktilní čidlo, piny pro vysílání a příjem dat (RX, TX) a servomotory.

```

#include <Servo.h>
#include <SoftwareSerial.h>
#define FSR_PIN A0          //Taktilní čidlo

SoftwareSerial BTserial(10, 11);      //TX,RX

Servo myservo1;                //malý prst
Servo myservo2;                //bezejmenný prst
Servo myservo3;                //střední prst
Servo myservo4;                //ukazovací prst
Servo myservo5;                //servo v lokti

char value;                    // přijatý příkaz z telefonu
int U_out;                     // výstupní napětí
float force;                   // síla ztlačení
int curr1, curr2, curr3, curr4, curr5;
int angle1, angle2, angle3, angle4, angle5;

```

Obrázek č. 15 — Definování proměnných.

Na obrázku číslo 15 je vidět, že jsem použil 4 serva na prsty a jedno servo na pohyb lokte. Na dlani jsou pouze 4 serva, protože velký palec je vyroben trochu křivě. Když se pohybují všechny prsty, palec občas překáží zpětnému pohybu ukazováčku. Také má příliš nízký vliv na určení síly stlačení, proto ho nemělo cenu používat.

Proměnné angle1, angle2 atd. jsou použity pro inkrementace úhlu pro každý motor v cyklu „for“. Proměnné curr1, curr2 jsem použil, aby například po úchopu míčku přestaly úhly inkrementovat a aby uvolnění ruky začínalo z aktuálního úhlu.

Lanka mezi špičkami prstů a servomotory jsou z oceli, proto po několika špatně nastavených úhlech lanka ztrácí svou kvalitu. Proto jsem nastavil konstantní minima a maxima pro úhly, které jsou pro každé servo různé.

```

const int MIN_ANGLE1 = 160;
const int MIN_ANGLE2 = 80;
const int MIN_ANGLE3 = 0;
const int MIN_ANGLE4 = 45;
const int MIN_ANGLE5 = 180;

const int MAX_ANGLE1 = 90;
const int MAX_ANGLE2 = 20;
const int MAX_ANGLE3 = 75;
const int MAX_ANGLE4 = 125;
const int MAX_ANGLE5 = 100;

```

Obrázek č.16 — Nastavení základních úhlů.

Servomotory naneštěstí občas mění úhel při kompilaci, proto je nutné hned na začátku nastavit pro každé servo své minimum. V mém případě je to `MIN_ANGLE1` atd.

Následujícím krokem je hlavní program, který bude řídit ruku. Při stisknutí tlačítka v aplikaci, telefon pošle určitá data do Arduina. Proto v metodě `loop` bude podmíněný příkaz, který bude kontrolovat, zda jsou v portu data.

```
void loop() {
  if (BTserial.available() > 0) {
    value = BTserial.read();
    BTserial.println(value);

    switch (value) {
      case 'c':
        clench_hand();
        break;
      case 'o':
        unclench_hand();
        break;
      case 'l':
        unclench_arm();
        break;
      case 'r':
        clench_arm();
    }
  }
}
```

Obrázek č.17 — Přepínač mezi návěstmi.

Jestli na port dorazí příkaz, tj. když podmínka `BTserial.available() > 0` bude vyplněna, poté všechny načtené příkazy se budou zapisovat do proměnné „value“ typu „char“.

Operace, která bude provedena záleží na přijatém příkazu. V přepínači `switch` jsou 4 návěští. Návěští „c“ znamená „close“ tj. stlačení dlaně. V návěští 'o' se provádí naopak uvolnění dlaně. Ostatní dvě návěští jsou pro pohyb lokte doleva a doprava.

Pro pohyb prstů jsem použil jednoduchý cyklus „for“. Po dosažení návěští 'c' náš program skočí ihned do metody „`clench_hand()`“. V této metodě probíhá inkrementace každého serva na určitý úhel, který byl nadefinován na začátku jako maxima. Mezi každou inkrementací program skočí do metody „`calculate_force()`“. Je to metoda pro určení síly stlačení, stručný popis kódu je ukázán v podkapitole 4.4. Metoda je typu `float`, která na konci metody vrátí hodnotu „force“ pomocí příkazu „`return force;`“.

```

void clench_hand() {

    for (angle1 = MIN_ANGLE1, angle2 = MIN_ANGLE2, angle3 = MIN_ANGLE3, angle4 = MIN_ANGLE4;
    angle1 >= MAX_ANGLE1 && angle2 >= MAX_ANGLE2 && angle3 <= MAX_ANGLE3 && angle4 <= MAX_ANGLE4;) {
        myservo2.write(angle2);
        myservo1.write(angle1);
        myservo3.write(angle3);
        myservo4.write(angle4);

        force = calculate_force();
        if (force < 5) {
            angle1--;
            angle2--;
            angle3++;
            angle4++;
        }

        if (force >= 5) {
            break;
        }
        delay(50);
    }
}

```

Obrázek č.18 — Kód odpovídající za zatnutí pěsti.

Podmínka „if (force <5)” znamená, že program vždycky porovnává vrácenou hodnotu force s nastavenou. Když je force menší než 5 N, servomotory se otočí do svého maxima. V opačném případě, resp. když je „force >= 5”, tak se motory zastaví. Zastavení motoru je realizováno pomocí příkazu „break”, který přeruší celý cyklus. Metoda „unclench_hand()” má stejný princip, ale liší se tím, že nemá kontrolu síly. Protože to pro uvolnění ruky nemá cenu určovat.

Ostatní dvě návěští jsou pro pohyb lokte. Zde jsem také použil jednoduchý cyklus „for”. Popis kódu je zobrazen na obrázku č. 19. Slovo „delay” znamená zdržení 50 ms po každé inkrementaci.

```

void unclench_arm() {

    for (angle5 = MIN_ANGLE5; angle5 >= MAX_ANGLE5;) {
        myservo5.write(angle5);
        angle5--;
        delay(50);
    }
}

void clench_arm() {
    for (angle5 = MAX_ANGLE5; angle5 <= MIN_ANGLE5;) {
        myservo5.write(angle5);
        angle5++;
        delay(50);
    }
}

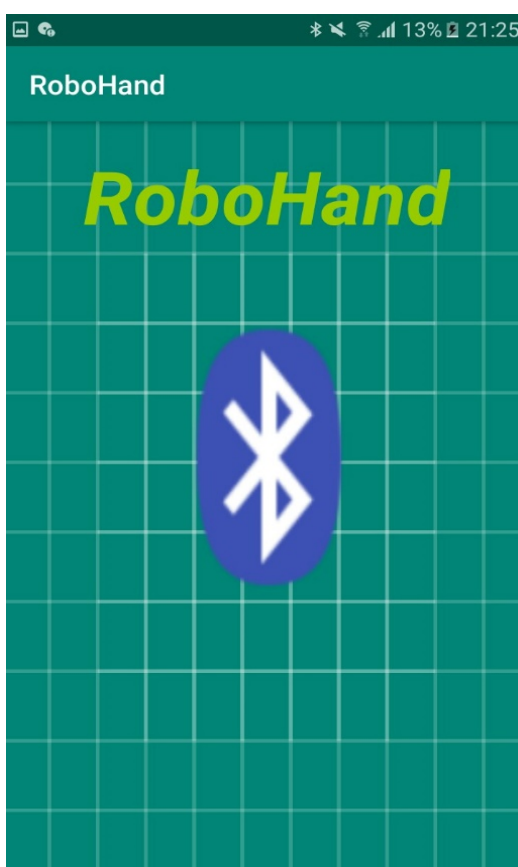
```

Obrázek č.19 — Pohyb lokte

5.2 Vytvoření mobilní aplikace

Pro napsání programu jsem použil Android Studio. Důvodem použití je, že má více zjednodušené rozhraní. Dalším z příčin, proč jsem vybral Android Studio je, že prostředí má emulátor telefonu. Dá se vybrat typ prošívání a lze nastavit rozlišení displeje a rozměry úhlopříčky. Výhodou je že při kompilaci do telefonu budou nastavené parametry stejné.

Založil jsem nový projekt s názvem „RoboHand“. Na začátku jsem vytvořil soubor MainActivity, který odpovídá za první stránku po spuštění programu. Na stránku jsem umístil název aplikace a hlavní obrazové tlačítko „Bluetooth“. Na obrázcích níže je ukázáno, jak by stránka vypadala, a je popsán i její stručný popis kódu.



Obrázek č. 20 — První stránka aplikace

Metoda onCreate je důležitou metodou v souboru MainActivity. Vytvoří se automaticky po vytvoření projektu. Bez této metody program fungovat nebude. Po stisknutí tlačítka přejdeme na stránku, ve které se nás aplikace zeptá, zda chceme zapnout Bluetooth v telefonu.

```

public class MainActivity extends AppCompatActivity {
    private ImageButton imageButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        imageButton = findViewById(R.id.imageButtonBT);
    }

    public void nextActivity(View view) {
        Intent intent = new Intent( packageContext: this, ControlScene.class);
        startActivity(intent);
    }
}

```

Obrázek č. 21 — Stručný popis kódu.

Tímto jsem vytvořil jen polovinu stránky. Druhá polovina spočívá v tom, že ve složce „layout“ je nutné vytvořit soubor typu XML (grafický návrhář), ve které lze vytvořit uživatelské rozhraní. To má název „návrhář uživatelského rozhraní“. V tomto souboru lze nastavit polohu tlačítek, textu, pozadí a atd. Pro stránku MainActivity bude mít XML soubor tvar:

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/ic_launcher_background">

```

Tato část odpovídá za pozadí, tzv. „background“.

```

<ImageButton
android:id="@+id/imageButtonBT"
android:layout_width="165dp"
android:layout_height="223dp"
android:layout_marginBottom="302dp"
android:background="@mipmap/bluetooth"
android:contentDescription="@string/desc"
android:onClick="nextActivity"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView" />

```

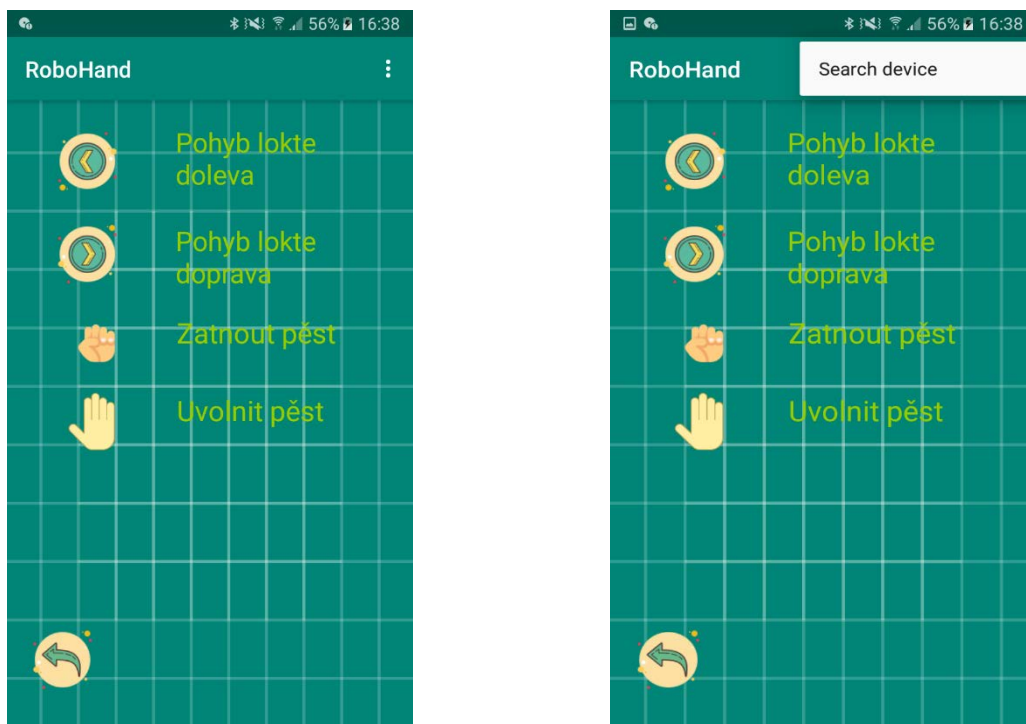
V této části jsou popsány parametry obrazového tlačítka Bluetooth. Řádek `android:background` ukazuje, že obrázek se nachází ve složce `mipmap`. Řádek `android:onClick` znamená, že po stisknutí tlačítka přejdeme na další stránku. Funkce pro „nextActivity“ je popsána v souboru MainActivity.

Na druhou stránku jsem umístil 4 tlačítka pro řízení dlaně a lokte, jedno tlačítko pro vrácení zpět a kontextové menu, sloužící pro vyhledávání zařízení. Na obrázku č. 22 jsem ukázal stručný popis přepínače „switch“ napsané v Arduino IDE. Pro řízení ruky je potřeba mít stejné příkazy, které přijímá Arduino pomocí Bluetooth modulu. Proto jsem v souboru ControlScene vytvořil třídu onTouchListener, ve které jsem napsal funkce přepínačů „switch“ pro posílání příkazů.

```
private final View.OnTouchListener touchListener =new View.OnTouchListener() {
    @SuppressWarnings("ClickableViewAccessibility")
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        String prikaz="";
        switch (event.getAction()){
            case MotionEvent.ACTION_DOWN:
                //Tlačítko pro pohyb lokte doleva
                if(v.equals(leftImageButton)){
                    isLeftImageButton=!isLeftImageButton;
                    prikaz = isLeftImageButton ? "l" : "s";
                    Log.d(TAG, msg: "OnClick: isLeftImageButton = " + isLeftImageButton);
                }
                //Tlačítko pro pohyb lokte doprava
                if (v.equals(rightImageButton)) {
                    isRightImageButton = !isRightImageButton;
                    prikaz = isRightImageButton ? "r" : "s";
                    Log.d(TAG, msg: "OnClick: isRightImageButton = " + isRightImageButton);
                }
                //Tlačítko pro zatnutí dlaně
                if (v.equals(clenchImageButton)) {
                    isClenchImageButton = !isClenchImageButton;
                    prikaz = isClenchImageButton ? "c" : "s";
                    Log.d(TAG, msg: "OnClick: isClenchImageButton = " + isClenchImageButton);
                }
                //Tlačítko pro uvolnění
                if (v.equals(unclenchImageButton)) {
                    isUnclenchImageButton = !isUnclenchImageButton;
                    prikaz = isUnclenchImageButton ? "o" : "s";
                    Log.d(TAG, msg: "OnClick: isDownImageButton = " + isUnclenchImageButton);
                }
                setMessage(prikaz);
                return true;
            }
        }
    }
}
```

Obrázek č. 22 — Popis přepínače switch.

Jak jsem již psal, třetí stránka bude obsahovat kontextové menu. Kontextové menu jsem použil pro vyhledávání zařízení. Kontextové menu obsahuje pouze jedno tlačítko (Search device). Po stisknutí tlačítka začne aplikace hledat Bluetooth zařízení v jejím okolí. Na obrázku č. 23 je ukázáno, kde je kontextové menu umístěno.



Obrázek č. 23 — Volání kontextového menu.

Dalším z podcílů této části práce je vysílání dat z Arduina do telefonu. V mém případě bych měl posílat stav aktuální hodnoty síly v Newtonech. Pro vysílání dat z telefonu byla použita knihovna *java.io.OutputStream*. Pro příjem dat je nutné použít knihovnu příjmu dat *java.io.InputStream*. Stručný popis kódu pro příjem dat je zobrazen na obrázku č. 24.

```

public void beginListenForData()
{
    final Handler handler = new Handler();
    stopThread = false;
    buffer = new byte[1024];
    Thread thread = new Thread((Runnable) () → {
        while(!Thread.currentThread().isInterrupted() && !stopThread)
        {
            try
            {
                int byteCount = inputStream.available();
                if(byteCount > 0)
                {
                    byte[] rawBytes = new byte[byteCount];
                    inputStream.read(rawBytes);
                    final String string=new String(rawBytes, charsetName: "UTF-8");
                    handler.post(() → {
                        textArduino.append(string);
                    });
                }
            }
            catch (IOException ex)
            {
                stopThread = true;
            }
        }
    });
}

```

Obrázek č. 24 — Část kódu odpovídající za příjem dat.

Funkci „*Thread*“ jsem použil pro založení nového toku informací. Použil jsem ji proto, že Android nemůže používat jeden tok pro vysílání a příjem dat.

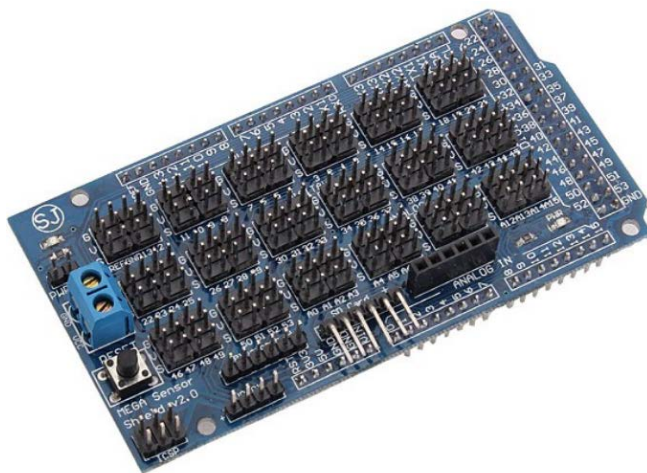
Pomocí grafického návrháře jsem přidal textové pole pro zobrazování aktuální hodnoty síly. Poté jsem textové pole propojil s kódem ve funkci „*beginListenForData()*“. Výsledkem jsou získaná data, která se zobrazují v reálném čase.



Obrázek č. 25 — Zobrazení aktuální síly v reálném čase.

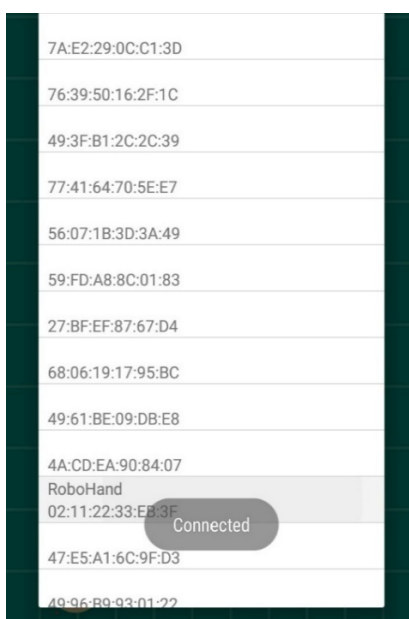
6. Testování funkčnosti aplikace a kódu

Po vytvoření programu byl následujícím krokem ověření, zda bud moci řídit ruku pomocí aplikace. Na začátku jsem zapojil všechny servomotory do Arduina, taktilní čidlo na analogový výstup A0, a ještě jsem připojil BT modul.



Obrázek č.26 — Arduino Mega senzor shield [18].

Servomotory potřebují velký proud, ale Arduino může vytvořit maximálně 150 mA, proto je nutné zapojit servomotory do vnějšího napájecího zdroje. Arduino Mega byl rozšířen pomocí Arduino Mega sensor shield (Obrázek č. 26). Poté byl shield připojen do zdroje. Po nahrání celého kódu a zapnutí zdroje jsem zkontroloval, zda je BT spuštěno. Dále jsem se k němu pokusil připojit z telefonu. Výsledek pokusu je zobrazován níže.



Obrázek č. 27 — Připojení k BT.

Po úspěšném připojení k BT už jen stačí pomocí tlačítek řídit pohyb ruky. Na začátku jsem zkusil pohybovat loktem ruky doleva, což fungovalo přesně dle očekávání. Po stisknutí tlačítka doprava jsem narazil na jeden problém. Ruka se chtěla vracet, na původní pozici, ale to se jí nedařilo. Bylo to způsobeno tím, že loket byl příliš těžký, a moment v lokti servomotoru malý, takže nestačil ke zvednutí ruky zpět. Při dodání síly se ruka zvedala zpět bez problémů.

Poté jsem stisknutím tlačítka zkusil zatnout dlaň v pěst, a zastavit ji v nějakém úhlu pomocí taktilního čidla. Výsledek odpovídal mým očekáváním, což znamená že se ruka pohybovala mírně a plynule. Uvolnění pěsti také funguje správně. Následující kapitolu budu věnovat návrhům pro zlepšení tohoto úkolu.

7. Návrhy pro zlepšení a použití

Jak jsem již psal v kapitole č. 4, navrhl bych výměnu čidla FSR 400 za FSR 400 Short, nebo za čidla Short jiných modelů. Důvodem této mnou navrhované změny je to, že při každém zatnutí pěsti čidlo začalo ztrácet svou kvalitu.

Za druhé by bylo vhodné namontovat čidlo na každý prst zvlášť. Tímto bychom zpřesnili velikosti měřených sil. Jako další návrh na zlepšení je natažení gumové rukavice na ruku, aby nedocházela k tak rychlému poškozování čidla.

Poté bych vyměnil servomotor v lokti za jiný, více výkonnější servomotor. Kvůli tomu, aby se ruka mohla zvedat sama, bez dalšího lidského zásahu.

V rámci této práce jsem pracoval pouze s robotickou rukou. Do budoucna, lze mnou vytvořenou aplikaci rozšířit pro řízení celého robota. Stačí přidat potřebný počet tlačítek, a dopsat kód pro ostatní návěští. Jako zjednodušení čitelnosti kódu, bych navrhoval přidat popis, který se objeví po stisknutí tlačítka, a napověděl by nám jakou funkci skrývá, nebo jaký příkaz posílá.

Síla, po které se ruka přestane pohybovat je nadefinována v kódu, resp. je statická pro všechny materiály. Nevýhodou je, že nastavený limit je pro křehké materiály dosti kriticky. Pro řešení toho problému lze přidat funkce v aplikaci pro nastavování síly, která je potřebná pro zastavení pohybu ruky.

8. Závěr

Cílem této bakalářské práce bylo vytvoření mobilní aplikace pro řízení ruky robota, která byla vytvořena a obhájena jako bakalářská práce pana Jakuba Vaňka, a která byla osazena taktilním čidlem.

Ve své práci jsem nejprve rešerší shrnul informace o taktilních čidlech, vývojových prostředích a možnostech bezdrátového řízení. Po shrnutí všech informací jsem se rozhodl použít Bluetooth modul HC-05 pro bezdrátové komunikace. Jako hardware jsem použil jednodeskový počítač Arduino Mega2560. Jelikož je Arduino otevřená elektronická platforma, použil jsem i vývojové prostředí Arduino IDE. Po srovnání všech platforem pro tvorbu aplikací, padla moje volba na Android Studio od firmy *Google*.

Během praktické části, která je zaměřena na umístění čidla na robotickou ruku a vytvoření vlastní aplikace jsem se potýkal s různými problémy. Na první jsem přišel při kompilaci jednoduchého kódu. Problém spočíval v tom, že některé servomotory uvnitř byly spálené a některé ocelová lanka roztrhaná. Po výměně všech poškozených součástí za nové se mi povedlo ruku zprovoznit.

Dalším důležitým bodem bylo otestování funkčnosti taktilního čidla a BT modulu. Při testování taktilního čidla proběhlo vše v pořádku, ale testování BT modulu neproběhlo několikrát správně, a trvalo mi dlouhou dobu problém vyřešit. Bylo to nejspíše zapříčiněno tím, že jsem si v teoretické části spletl typy modulů BT. Po opakovaném shrnutí informací o BT modulech problém vymizel.

Aby vše proběhlo korektně, rozhodl jsem rozdělit praktickou část na řízení ruky a vytvoření vlastní aplikace. Řízení ruky bylo na začátku ovládáno pomocí tlačítek. Po dosažení cíle této části jsem začal tvořit první vlastní aplikaci.

Jelikož jsem byl v této oblasti úplným začátečníkem, bylo nutné najít a nastudovat co nejvíce zdrojů, kde bylo pochopitelně popsáno, jak správně začít programovat. Po získání základních informací o Android Studio jsem se pokusil otestovat své znalosti v praxi. Výsledkem byla aplikace „*RoboHand*“, která byla znázorněna na obrázcích výše.

Přínos mé bakalářské práce vidím hlavně ve vytvoření aplikace „*RoboHand*“, která bude moci najít své využití při ovládání nejen robotických částí, jakou byla v našem případě ruka, ale i celého robota. Toho budeme moci využít v průmyslu, a i ve stále se rozvíjejícím oboru biomedicíny.

Literatura

- [1] Asutpp.ru. *Princip činnosti taktilních čidel* [online]. 2015 [cit. 2019-04-11]. Dostupné z:
<https://www.asutpp.ru/tenzodatchik.html>
- [2] Cxem.net. *Taktilní čidlo FSR402* [online]. [cit. 2019-04-15]. Dostupné z:
<http://cxem.net/arduino/arduino217.php>
- [3] Adafruit.com. *Force Sensitive Resistor (FSR)* [online]. 2012 [cit. 2019-04-15]. Dostupné z:
<https://learn.adafruit.com/force-sensitive-resistor-fsr/overview>
- [4] Electrosam.ru *Taktilní čidla. Princip činnosti a typy.* [online]. 2019 [cit. 2019-04-16]. Dostupné z:
<https://electrosam.ru/glavnaja/jelektrooborudovanie/ustrojstva/datchiki-davleniia/>
- [5] Interlink ELECTRONICS. *FSR 400 Series Data Sheet* [online]. [cit. 2019-04-20]. Dostupné z:
https://cdn2.hubspot.net/hubfs/3899023/Interlinkelectronics%20November2017/Docs/Datasheet__FSR.pdf
- [6] VOLF, Jaromír. *Taktilní senzory a jejich využití v robotice a biomechanice* [online]. V Praze: České vysoké učení technické, 2008 [cit. 2019-04-20]. ISBN 978-80-01-04082-9. Dostupné z:
<https://portal.cvut.cz/wp-content/uploads/2017/04/PP2008-02-Volf.pdf>
- [7] Sps-ko.cz. *Průmyslové roboty* [online]. 2018 [cit. 2019-04-21]. Dostupné z:
<http://www.sps-ko.cz>
- [8] Arduino-diy.com. *Taktilní senzor a Arduino* [online]. 2016 [cit. 2016-04-22]. Dostupné z:
<http://arduino-diy.com/arduino-rezistivnyy-datchik-davleniya>
- [9] BLUM, Jeremy. *Exploring Arduino: Tools and Techniques for Engineering Wizardry* [online]. Indianapolis, IN: Wiley, 2013 [cit. 2019-04-25]. ISBN 978-1-118-54936-0. Dostupné z:
<https://www.academia.edu>
- [10] ZBYŠEK, Voda. *Průvodce světem Arduina* [online]. 2. vyd. Bučovice: Martin Stříž, 2017 [cit. 2019-04-25]. ISBN 978-80-87106-93-8. Dostupné z:
<http://robotikabrno.cz>
- [11] Ardiono.cc. *Arduino Mega 2560* [online] [cit. 2019-04-25]. Dostupné z:
<https://store.arduino.cc/arduino-mega-2560-rev3>

- [12] Spezial.cz. *BLUETOOTH SERIAL PORT ADAPTER AT COMMANDS* [online]. 2013 [cit. 2019-05-04]. Dostupné z: http://www.spezial.cz/pdf/Serial_Port_Adapter_AT_Commands.pdf
- [13] Arduino-shop.cz. *Bluetooth HC-06* [online]. [cit. 2019-05-04]. Dostupné z: <https://arduino-shop.cz/arduino>
- [14] HORTON, John. *Android programming for beginners*, Packt Publishing, 2015, [cit. 2019-05-10] ISBN-13: 978-1785883262. Dostupné z: <https://x.coe.phuket.psu.ac.th>
- [15] DiMarzio, J.F. *Android Programming with Android Studio*. [online]. 4 vyd. 2017, [cit. 2019-05-12] ISBN 978-1-118-70559-9. Dostupné z: http://solutionsproj.net/software/Beginning_Android_Programming_with_Android_Studio.pdf
- [16] VANĚK, Jakub. *Návrh humanoidní ruky robota*. [online]. 2018 [cit. 2019-05-13] Dostupné z: <https://dspace.cvut.cz>
- [17] Android.com *Android Studio*. [online] [cit. 2019-05-12] Dostupné z: <https://developer.android.com/studio>
- [18] Arduino-shop.cz. *Arduino Mega senzor shield* [online]. [cit. 2019-05-04]. Dostupné z: <https://arduino-shop.cz/>
- [19] GERBER, Adam a CLIFTON, Craig. *Learn Android Studio*. [online] 2015, ISBN 978-1-4302-6602-0. Dostupné z: http://barbra-coco.dyndns.org/student/learning_android_studio.pdf
- [20] Livetyping.com. *Na čem se píše Android aplikace*. [online] 2018, Dostupné z: <https://livetyping.com/ru/blog/na-chem-pishut-prilozhenija-pod-android>
- [21] Auklet.io. *C/C++ Solution Identification With Auklet*. [online] Dostupné z: <https://www.auklet.io/edge-c-c/>
- [22] Zemicusa.info. *Víte kde jsou používány taktilní čidla?* [online] 2015, [cit. 2019-06-10] Dostupné z: <http://zemicusa.info/poleznye-stati/209-znaete-li-vy-gde-primenyayutsya-tenzodatchiki>
- [23] Studfiles.net. *Taktilní čidla*. [online] 2014, [cit. 2019-06-10] Dostupné z: <https://studfiles.net/preview/774681/page:9/>
- [24] Bezprovodoff.ru. *Co to je síť Bluetooth a jak funguje*. [online] 2016, [cit. 2019-06-10] Dostupné z: <http://bezprovodoff.com/seti/vidy/chto-takoe-bluetooth.html>
- [25] Amperka.ru. *Wireless Fidelity*. [online] [cit. 2019-06-10] Dostupné z: <https://wiki.amperka.ru/>