



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

DEPARTMENT OF INTELLIGENT SYSTEMS

**VÝUKOVÝ TUTORIAL PRO OVLÁDÁNÍ ROBOTA PO-  
MOCÍ MOBILU**

EDUCATIONAL TUTORIAL FOR SMARTPHONE ROBOT CONTROL

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MATEJ TUREK**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. JAROSLAV ROZMAN, Ph.D.**

BRNO 2020

## Zadání bakalářské práce



Student: **Turek Matej**  
Program: Informační technologie  
Název: **Výukový tutorial pro ovládání robota pomocí mobilu**  
**Educational Tutorial for Smartphone Robot Control**  
Kategorie: Softwarové inženýrství

### Zadání:

1. Nastudujte tvorbu jednoduchých aplikací pro mobilní telefony se systémem Android a výukové tutoriály.
2. Navrhněte výukový tutorial, který vysvětlí, jak si vytvořit program, který umožní řízení robota (jízda vpřed a vzad a zatáčení) a čtení hodnot z různých senzorů. Počítejte s možností propojení mobilního telefonu a např. Arduina, Raspberry Pi nebo jiného. Data z robota mohou být buď simulovaná nebo z reálného robota. Tutorial navrhněte tak, aby se dalo vybrat mezi různou obtížností/délkou/hloubkou problematiky.
3. Podle navrženého tutorialu vytvořte program pro Android a to jak kompletně hotový, tak také s vymazanými částmi, které budou uživatelé doplňovat. Dále vytvořte slajdy, které se použijí pro prezentaci při výuce. Slajdů vytvořte více verzí podle času nutného pro projití tutorialem a udělejte je jak v českém, tak i v anglickém jazyce.
4. Vytvořený tutorial otestujte na skupině uživatelů. Na základě zpětné vazby navrhněte případná vylepšení.

### Literatura:

- Internetové tutoriály pro Arduino

Pro udělení zápočtu za první semestr je požadováno:

- První dva body zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Rozman Jaroslav, Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2019

Datum odevzdání: 28. května 2020

Datum schválení: 31. října 2019

## Abstrakt

Táto práca popisuje vytvorenie návodov pre ovládanie robota postaveného na platforme Android Mega 2560 pomocou smartfónu s operačným systémom Android. K robotovi sú pripojené senzory, ktoré vysielajú informácie o stave robota. Komunikácia prebieha formou Bluetooth modulu HC-05, na základe ktorej si smartfón a robot vymieňajú dáta. Smartfón tieto dáta spracováva a posiela pokyny späť do robota. Jednotlivé návody sa líšia v obtiažnosti dopĺňovania pripravenej kostry programu, ktorý je napísaný v jazyku Java. Tieto návody je možné využiť pre výučbu študentov s minimálnymi skúsenosťami, mierne pokročilejšími alebo pokročilejšími skúsenosťami s programovaním.

## Abstract

This thesis describes making of tutorials to control robot based on platform Arduino Mega 2560 with smartphone based on operating system Android. Robot does have sensors that sends information about its state. Communication is created with Bluetooth module HC-05 that allows that to exchange data. Then smartphone process the data and send instructions to the robot. The types of tutorials differs in difficulty and there is prepared unfinished program which tutorials complete. Program is written in Java. Tutorials could be used for teaching purpose with minimal, slightly advanced and advanced experience with programming.

## Klíčová slova

Arduino Mega 2560, Android, Android Studio, Java, Bluetooth, HC-05, robot, návod

## Keywords

Arduino Mega 2560, Android, Android Studio, Java, Bluetooth, HC-05, robot, tutorial

## Citace

TUREK, Matej. *Výukový tutorial pro ovládání robota pomocí mobilu*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Rozman, Ph.D.

# Výukový tutorial pro ovládání robota pomocí mobilu

## Prohlášení

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Jaroslava Rozmana, Ph.D. Ďalšie informácie mi poskytl... Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....

Matej Turek  
27. května 2020

## Poděkování

Chcel by som úprimne poďakovať vedúcemu mojej práce pánovi Ing. Jaroslavovi Rozmanovi, Ph.D. za konzultácie, za podporu a ochotu poradiť pri akýchkoľvek otázkach a problémoch, ktoré nastali počas tvorby tejto práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Návody</b>	<b>4</b>
2.1	Typy návodov . . . . .	4
2.2	Tvorba aplikácie . . . . .	11
<b>3</b>	<b>Návrh návodu</b>	<b>13</b>
3.1	Najjednoduchší návod . . . . .	13
3.2	Pokročilý návod . . . . .	14
3.3	Najťažší návod . . . . .	14
3.4	Ľudský faktor . . . . .	14
3.5	Návrh aplikácie . . . . .	15
<b>4</b>	<b>Referenčné riešenie</b>	<b>17</b>
4.1	Programovanie . . . . .	17
4.2	Zapojenie Arduina k modulu a senzorum . . . . .	18
4.2.1	Bluetooth HC-05 . . . . .	18
4.2.2	Senzory vzdialenosti . . . . .	19
4.2.3	IMU . . . . .	20
4.2.4	Motor . . . . .	21
4.3	Program Arduina . . . . .	23
4.4	Program Androidu . . . . .	24
4.4.1	DevicesActivity . . . . .	25
4.4.2	ControlActivity . . . . .	26
4.4.3	Spracovanie dát . . . . .	26
4.4.4	Užívateľské rozhranie . . . . .	26
<b>5</b>	<b>Návod</b>	<b>28</b>
5.1	Začiatočníci . . . . .	28
5.1.1	Vývojové prostredie . . . . .	28
5.1.2	Projektová štruktúra . . . . .	29
5.1.3	Tvorba jednoduchého programu . . . . .	30
5.1.4	Úvod do ovládania robota . . . . .	31
5.1.5	Spracovávanie dát a prezentácia . . . . .	32
5.1.6	Záver . . . . .	32
5.2	Pokročilí . . . . .	32
5.2.1	Vývojové prostredie a program . . . . .	32
5.2.2	Úvod do ovládania robota . . . . .	32

5.2.3	Vytvorenie komunikácie s robotom . . . . .	32
5.2.4	Spracovávanie dát . . . . .	33
5.2.5	Záver . . . . .	33
5.3	Skúsení . . . . .	33
5.3.1	Robot . . . . .	34
5.3.2	Úloha vytvorenie užívateľského rozhrania . . . . .	34
5.3.3	Úloha spárovanie zariadení . . . . .	35
5.3.4	Vytvorenie komunikácie . . . . .	35
5.3.5	Spracovávanie dát a reprezentácia . . . . .	36
5.3.6	Záver . . . . .	36
<b>6</b>	<b>Testovanie</b>	<b>38</b>
6.1	Začiatočníci . . . . .	38
6.2	Pokročilí . . . . .	38
6.3	Skúsení . . . . .	39
6.4	Zhodnotenie . . . . .	39
	<b>Literatura</b>	<b>40</b>
	<b>A Návody</b>	<b>42</b>
	<b>B Program Android</b>	<b>43</b>
	<b>C Vzorový program Arduino</b>	<b>44</b>

# Kapitola 1

## Úvod

Tato práca sa zaoberá vytvorením tutoriálu pre ovládanie robota pomocou smartfónu s operačným systémom Android. Práca je rozdelená na viacero častí. Prvé časti obsahujú témy týkajúce sa návodov a ich rozdelenie do typov. Pri každom type sú určené konkrétne vlastnosti a na záver je porovnanie. Taktiež obsahujú tvorbu aplikácií na smartfóny postavené na operačnom systéme Android. Ďalšou časťou sú návrhy na návody pre cieľové skupiny študentov a popisujú, čo všetko by tieto návody mali obsahovať. Nasledujúce časti obsahujú samotnú tvorbu aplikácie na ovládanie robota, ktorá bude slúžiť ako referenčné riešenie tohoto projektu bude výsledkom všetkých návodov spomínaných v predošlej kapitole. Posledná kapitola obsahuje konkrétne riešenie návodov a konkrétne riešenia, ktoré budú študenti počas výučby riešiť.

# Kapitola 2

## Návody

V tejto kapitole je popis a porovnanie návodov a prerekvizity k vytvoreniu aplikácie. Návody sa rozdeľujú do viacerých sekcií a to podľa formy. Internet je hlavným zdrojom týchto návodov. Medzi základné typy patria:

- Dokumentácia
- Blog
- Video
- Prezentácia

Na záver je zhodnotenie a porovnanie návodov. Následne je vybraný najvhodnejší návod pre tento projekt.

### 2.1 Typy návodov

#### Návod formou dokumentácie

Tento typ návodu rozoberá každú časť programovania jednotlivu v kapitolách, takže tento návod je rozsiahly. Začína od najjednoduchších vecí ako sú inštalácia a nastavenie vývojového prostredia (Android Studio), ďalej popisuje základnú štruktúru programu, popis tvorby užívateľského prostredia, príklad tzv. Hello World atď. Taktiež sú návody rozdelené do sekcií a tie do podsekcií, ako sú napríklad: základy a začiatky, zariadenia, užívateľské rozhranie, programovanie funkcionality, príklady, atď.

Zaujímavé je to z hľadiska toho, že čitateľ môže jednoducho preskočiť jednotlivé kapitoly a prejsť si konkrétny problém, ktorý ho zaujíma. Druhou možnosťou je prechádzanie si to systematicky od začiatku po koniec a teda od najjednoduchších vecí po najťažšie. Na ľavej strane obsahuje menu, v ktorom sa ľahko orientuje a urýchľuje to vyhľadávanie.

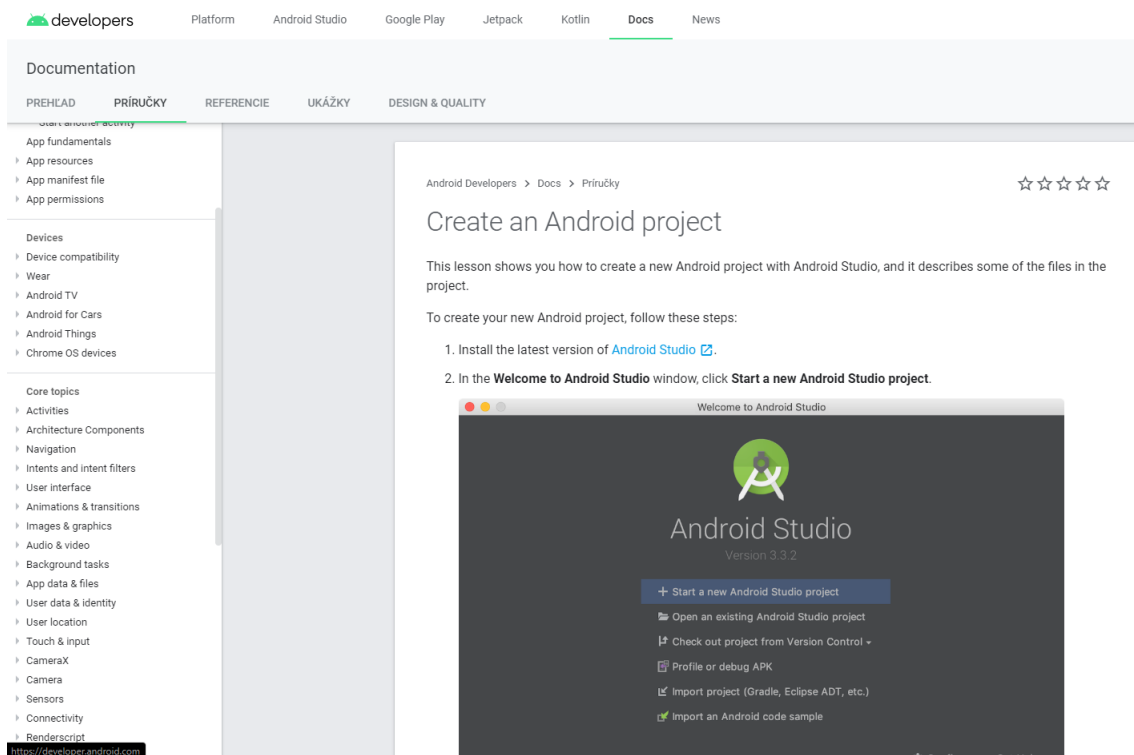
Obsahuje to množstvo príkladov kódov, názorné obrázky (napríklad pri tvorbe užívateľského rozhrania), tabuľky, schémy. Najviac obsahuje text, ktorým sa tento návod snaží vysvetliť danú tému. Toto všetko je možné skopírovať a využiť vo svojom projekte.

Jazyk v ktorom bývajú tieto dokumentácie napísané je zvyčajne angličtina.

Veľké množstvo iných tutoriálov odkazuje priamo na tieto stránky, ktoré slúžia ako referenčné. Tým pádom sú stále aj aktualizované, napríklad pri vydaní novej verzie operačného systému Android a to je najväčšia výhoda.

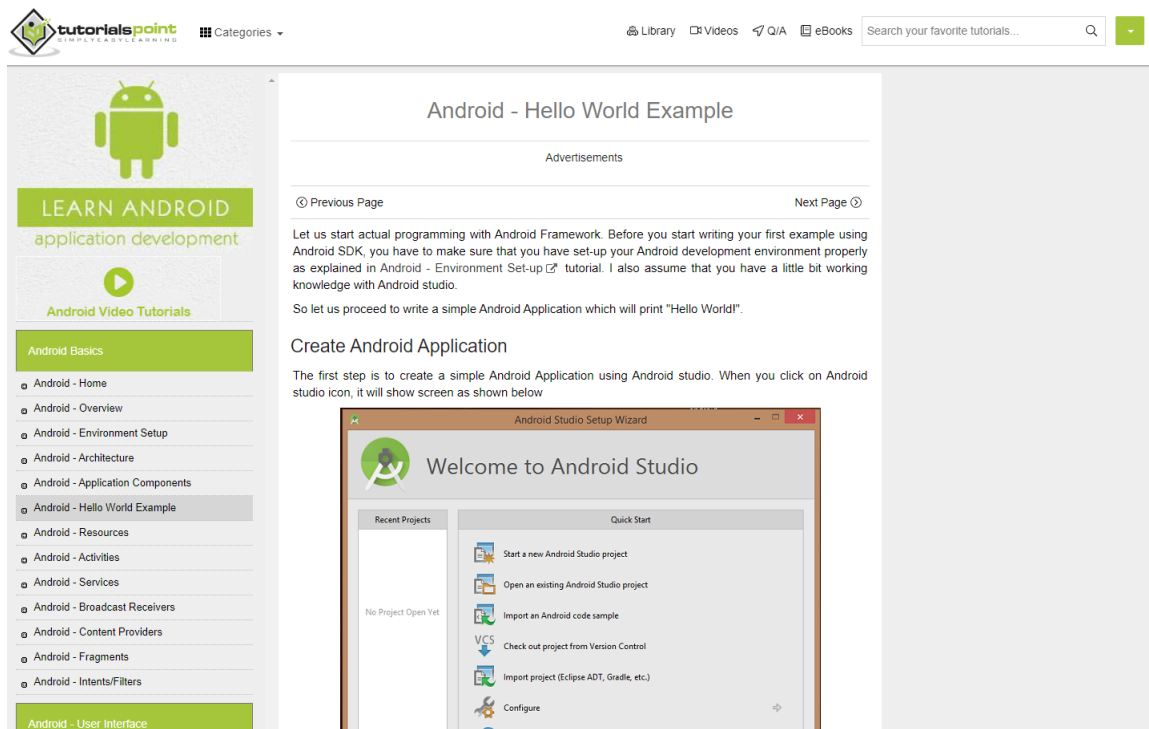


Ako typický príklad je [5] Android developers, na ktorej je aj jeden z týchto návodov, obsahuje priamo aj fórum, kontaktný formulár, odkazy na stiahnutie, novinky, ...



Obrázek 2.1: Návod formou dokumentácie 1

Ďalším príkladom je [16] Tutorialspoint. Táto stránka obsahuje množstvo takýchto návodov týkajúcich sa všeobecne programovania rôznych programovacích jazykov.

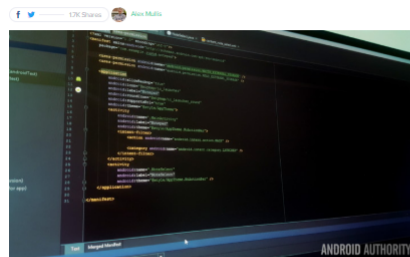


Obrázek 2.2: Návod formou dokumentácie 2

## Návod formou blogu

Tento typ návodu [13] [7] je zameraný predovšetkým na jednu konkrétnu tému. Je písaný na jednu dlhú stránku, ktorá obsahuje text, ale aj obrázky a aj kód. Zvolená obtiažnosť závisí od konkrétnej témy a jej spracovania.

## Android Studio tutorial for beginners



There are multiple ways to approach Android Development but by far the most official and powerful is to use Android Studio. This is the official IDE (Integrated Development Environment) for the Android platform, developed by Google and used to make the majority of the apps that you probably use on a daily basis.

[Read Next: Java tutorial for beginners](#)

Android Studio was first announced at a Google I/O conference in 2013 and was released to the general public in 2014 after various beta versions. Prior to its release, Android development was handled predominantly through Eclipse IDE, which is a more generic Java IDE that also supports numerous other programming languages.

Android Studio makes life significantly easier compared with non-specialist software, but it still has a little way to go before it can claim to be a completely intuitive and smooth experience. For complete beginners, there is an awful lot to learn here and much of the information available – even through official channels – is either out of date or too dense to make head or tails of.

In this post, we'll explain what Android Studio does in a little more detail and go over the basic functionality that you need to get started. I'll try and keep everything as easy as possible and hopefully this will serve as the first step on your journey to Android Development.

### So just what is Android Studio?

Those of you with no prior experience in coding may still be wondering precisely what Android Studio's role is when it comes to development... what is an IDE anyway?



Obrázek 2.3: Ukážka návodu formou blogu 1

Hodia sa väčšinou pre záujemcov o konkrétnu tému, takže od toho závisí cieľová skupina študentov. Taktiež sa hodia pre rôzne rýchle vyhľadávanie konkrétnej témy alebo konkrétneho problému, pretože sú krátke. Závisí to však od formy spracovania a vysvetlenia. Často bývajú chaotické alebo neprehľadné. Niektoré však obsahujú nové riešenia, ktoré je možné len ťažko nájsť, pretože prezentujú úplne nové prístupy a knižnice. Na druhú stranu, niektoré prezentujú vopred známu problematiku, ktorú je možné dohľadať vo veľa iných dostupných zdrojoch.

Pre tento projekt sú však nevhodné a ostatné typy často využívajú lepších a prehľadnejších spôsobov na riešenie.

## Android Studio for beginners, Part 1: Installation and setup

Install Android Studio and start your first Android project. Updated January 2020



Updated: January 2020.

For some years now it's been clear that [Android](#) is a force to be reckoned with in the mobile OS landscape. This Java-based technology has sparked a new [gold rush](#), with programmers competing to [make money from their mobile apps](#). Android jobs are also plentiful, as shown by a quick job search using [Indeed.com](#).

To be successful, Android developers need a good grasp of the [Java language](#) (or [Kotlin](#)), [Android APIs](#), and [Android application architecture](#). It's also important to use an appropriate and effective development environment. For many years, [Eclipse IDE with the ADT plugin](#) was the preferred platform for Android development. Today, it's [Android Studio](#).

Obrázek 2.4: Ukážka návodu formou blogu 2

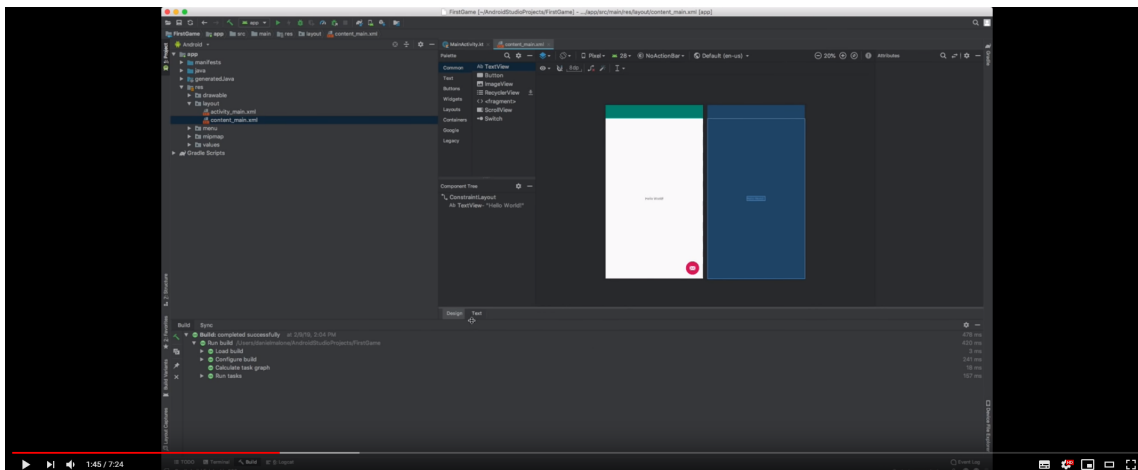
### Návod formou videa

Tento typ návodu [12] [6] je vhodný skôr pre jednotlivcov. Rozoberá viacero častí, postupne od najľahšej po najťažšiu. V prvých častiach sa vysvetľujú základné veci ako sú inštalácia vývojového prostredia a založenie projektu, základná schéma atď. Zvyčajne sa pracuje s tým istým projektom, ktorý je postupne obohacovaný o ďalšiu a ďalšiu funkcionality, až dokým sa nedostane do konečnej formy. Vzhľadom k tomu, že je tento typ sprevádzaný zvukom, je lepšie vysvetliteľný ako predchádzajúce. Umožňuje preskakovanie pasáží a naopak umožňuje aj vrátenie sa.

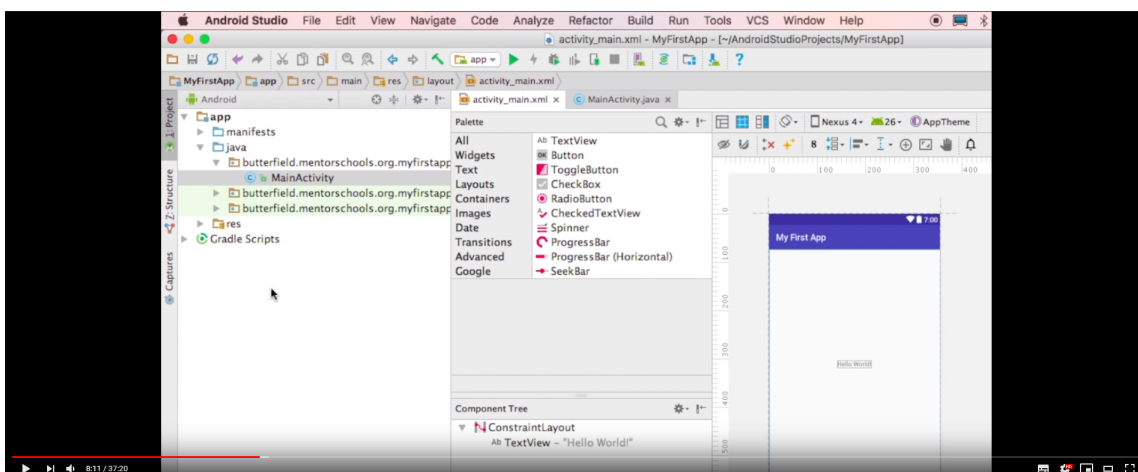
Tieto návody nevyžadujú žiadnu predchádzajúcu skúsenosť s programovaním, preto sú vhodné pre všetky typy študentov. Študenti taktiež vidia samotný kód a postupy tvorby programov, takže o to je to pre nich jednoduchšie.

Ako negatívum sa dá považovať riešenie problémov týkajúcich sa napríklad nepreložitelných kódov, problémov s vývojovým prostredím alebo zastaralými knižnicami. Vzhľadom k tomu, že tieto návody nie je možné udržiavať aktuálne, často sa stáva, že sú už nekompatibilné s novšími zariadeniami alebo naopak existujú novšie knižnice alebo spôsoby, ktoré je možné využiť, ale tento návod to neposkytuje.

Ako pozitívum je dostupnosť. Existuje veľké množstvo týchto návodov voľne dostupných. Najčastejší jazyk je angličtina, ale nie je ťažké nájsť s češtinou alebo so slovenčinou. Ďalším pozitívom je riešenie problémov. Často sa stáva, že sa vyskytne nejaký problém v programovaní a študent vidí a učí sa ako takýto problém riešiť.



Obrázek 2.5: Návod formou videa 1



Obrázek 2.6: Návod formou videa 2

## Návod formou prezentácie

Tento typ návodu [8] [15] je bežný na výučbách, kde je sprevádzaný prejavom vyučujúceho alebo cvičiaceho. Je vhodný pre relatívne malú skupinu študentov, pretože by to malo byť interaktívne. Od spracovania a prejavu závisí, aká je cieľová skupina pre tento typ návodu.

Sú veľmi vhodné na nejakú konkrétnu tému. Je možné ju rozoberať priamo od základov alebo prispôbiť sa podľa potrieb študentov. Lektor hrá dôležitú úlohu, pretože táto forma je interaktívna. Pri akomkoľvek probléme je pripravený ho riešiť, pretože lektor by mal byť na takéto situácie pripravený a mať s tým už skúsenosti.

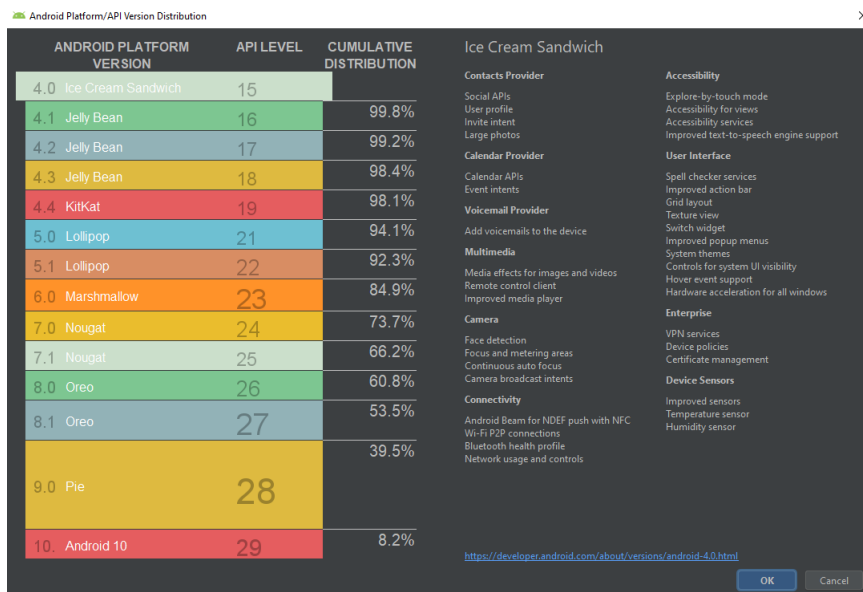
Ako nevýhodu je možné považovať skupinu študentov. Pretože to je skupina, je potrebné sa prispôbiť najpomalšiemu členovi tejto skupiny. To znamená, že ak sa vyskytne nejaký problém u jedného študenta, zvyšok študentov musia počkať na vyriešenie tohoto problému lektorom. Preto je nevyhnutné, aby skupina študentov mala podobné skúsenosti s programovaním a vedomosti, aby sa tento problém minimalizoval čo najviac. Aj keď pri



## 2.2 Tvorba aplikácie

Programovanie aplikácií na operačný systém Android je postavené na základe programovacieho jazyka Java alebo Kotlin. V tomto projekte bola použitá Java a to z toho dôvodu, pretože je podobnejšia niektorým programovacím jazykom, ako je napríklad C++, s ktorými už niektorí študenti majú skúsenosti.

Vzhľadom k tomu, že existuje viacero verzií operačného systému Android, je potrebné aby sa zohľadnil aj výber minimálnej verzie. Dôležitú úlohu hrá výber SDK (angl. Software Development Kit), resp. API (ang. Application Programming Interface), teda súbor nástrojov, ktoré umožňujú vytváranie softvéru. Pri výbere hrá dôležitú úlohu nasledujúca tabuľka:



Obrázek 2.9: Percentuálne rozloženie Android zariadení

Táto tabuľka ukazuje štatistiku používaných zariadení. Z nej vyplýva, že využitie od Androidu verzie 4.0 je viac ako 99%, preto nemá zmysel sa zaoberať nižšími verziami na úkor obmedzenia funkcionality. **Zvolená verzia je teda API level 15 a operačný systém Android 4.0 Ice Cream Sandwich**

Ako výukové vývojové prostredie je zvolená jasná voľba. Je to Android Studio [4], ktoré je voľne stiahnuteľné priamo na stránkach developerov. Je postavené na báze IntelliJ IDEA. Poskytuje možnosť emulátoru zariadenia, jednotnosť pre všetky zariadenia Android, možnosť aplikovať zmeny priamo za behu programu a mnoho ďalších. Je veľmi intuitívne a ľahké na ovládanie.

Prerekvizity pre tvorbu aplikácií sú nasledovné:

- Java development kit (JDK) verzia 5 a vyššie,
- Java Runtime Environment (JRE) 6
- Android Studio
- Android emulátor alebo Android zariadenie

Pre spustenie projektu je potrebné jedno z dvoch možných zariadení virtuálne a reálny mobil. Pre spustenie virtuálneho emulátoru je potrebné mať zapnutú virtualizáciu (VT-x). Pre spustenie s reálnym mobilom je potrebné, aby bol s operačným systémom Android akejkoľvek verzie, pripojený USB káblom a zapnuté USB ladenie v nastaveniach. Pre tento projekt sa však emulátor nehodí, preto je využité obyčajné Android zariadenie.

Po tom, ako je všetko pripravené (vývojové prostredie a aj koncové zariadenie), je možné aplikáciu preložiť a spustiť. Po spustení sa aplikácia nainštaluje a spustí na koncovom zariadení. Po spustení je možné aplikáciu ladiť, ale Android Studio umožňuje ladenie aj tesne po spustení po začiatku (to znamená už od prvého riadku kódu).



## Kapitola 3

# Návrh návodu

Ako sa uvádza v predchádzajúcej kapitole, návody budú vo forme prezentácie, budú tri a budú rozlíšené podľa obtiažnosti a to nasledujúcim spôsobom:

1. Najjednoduchší návod bude zameraný na tvorbu užívateľského rozhrania, ale bude obsahovať aj jednoduché dopĺňanie kódu.
2. Pokročilý bude kompromisom medzi najľahším a najťažším návodom. To znamená, že je zameraný aj na užívateľské rozhranie ale aj programovanie samotnej funkcionality (ktorá avšak bude oproti najťažšiemu návodu jednoduchšia).
3. Najťažší návod bude zameraný predovšetkým na funkcionality - prijímanie, spracovávanie a posielanie dát medzi smartfónom a robotom. A to z dôvodu, že sa predpokladá, že títo študenti majú už nejaké skúsenosti s tvorbou užívateľského rozhrania a bolo by to pre nich zbytočné.

Návody budú jak v češtine tak aj v angličtine.

Z pohľadu prezentácie budú pripravené dva súbory - pdf súbor s daným návodom a kostra programu, prípadne ďalšia literatúra. Kostrou programu je možné chápať pripravené funkcie, užívateľské rozhranie, ktoré nie je dokončené. Tým pádom je na študentoch, aby toto dokončili a výsledným produktom bude samotný projekt na ovládanie robota pomocou smartfónu. Kostra sa líši medzi jednotlivými kategóriami obtiažnosti. Je dôležité, aby chýbajúce časti boli minimálne pri najťažšom návode zamerané na tie časti kódu, ktoré vysvetľujú, základnú funkcionality programu.

### 3.1 Najjednoduchší návod

Vychádza sa z predpokladu, že títo študenti nemajú veľa skúseností s programovaním. Prvou témou návodu by malo byť porozumenie a spoznávanie vývojového prostredia Android Studio. To zahŕňa založenie projektu, preklad, spustenie a ladenie projektu, ovládacie prvky vývojového prostredia a prípadne základy programovacieho jazyka. Ďalším krokom je štruktúra projektu. To zahŕňa vysvetlenie štyroch dôležitých typov súborov - res, activity (a Java súbory), manifest a gradle. Tretím krokom by mala byť tvorba nejakého jednoduchého základného projektu, kde by si mohli vyskúšať jak tvorbu užívateľského rozhrania, ale aj zoznámiť sa so základnou funkcionality. Takým projektom by mohla byť napríklad jednoduchá kalkulačka. Tak by si spojili všetky skúsenosti a vedomosti z predošlých krokov a viedlo by ich k lepšiemu priebehu pri programovaní projektov v Androide. Poslednou

kapitolou návodu je samotný projekt pre ovládanie robota. Vzhľadom k tomu, že z prázdneho projektu by to bolo príliš náročné, je potrebné pripraviť nejakú kostru programu, kde budú mať tie pre nich najnáročnejšie veci už pripravené a môžu sa zaoberať vecami, ktoré sú jednoduchšie a primerané ich schopnostiam. To zahŕňa vytvorenie užívateľských prvkov ako sú napríklad tlačítka a textové políčka, ale aj jednoduché programovanie funkcionality ako je napríklad obsluha týchto užívateľských prvkov.

## 3.2 Pokročilý návod

Tento návod predpokladá už nejakú skúsenosť s programovaním a vývojovým prostredím. Tým pádom sa preskakuje zakladanie projektu, spustenie projektu a orientovanie sa vo vývojovom prostredí. Prvou kapitolou by mal byť popis štruktúry projektu. Štruktúra projektu môže byť zo začiatku mäťúca, preto je to potrebné vysvetliť, aby sa ľahšie v projekte orientovalo. Ďalšou časťou návodu je tvorba nejakého programu, kde sa kladie dôraz na nielen ovládanie prvkov v užívateľskom rozhraní, ale taktiež funkcionality na pozadí. Opäť sa môže využiť kalkulačka ako pri jednoduchom návode, ale napríklad obohatená o ďalšie operácie alebo so samostatným prístupom študentov. Poslednou kapitolou je podobne ako v predchádzajúcom návode samotný projekt. Rozdiel nastáva v doplnení kostry. Dôraz kladie nielen a prvky užívateľského rozhrania, ale aj na samotný kód funkcionality. Tu sa napríklad môže vyskytnúť do určitej miery samotná komunikácia cez Bluetooth, spracovávanie dát.

## 3.3 Najťažší návod

V tomto návode je neodmysliteľnou súčasťou skúsenosti s programovaním. Nemusí sa to konkrétne týkať daného projektu, teda programovacieho jazyku Java alebo vývojového prostredia Android Studio. Je však dôležité chápať základné štruktúry a princípy programovania, od ktorých sa budú odvíjať ďalšie postupy riešenia projektu. Prvým krokom by malo byť opäť vysvetlenie štruktúry projektov pre Android, ale trochu iným spôsobom ako v predchádzajúcich návodoch. To znamená detailnejšie a podrobnejšie. Možno je to vysvetliť na iných princípoch - napríklad návrhové vzory. Druhou a poslednou témou by malo byť samotný projekt. Keďže sa predpokladá, že študent má už nejaké skúsenosti s užívateľským rozhraním, je dôležité, aby sa pozornosť upriamila na funkcionality. Tu by sa malo postupne pracovať po jednotlivých krokoch. Najprv vyriešiť komunikáciu medzi smartfónom, ďalej spracovávať jednotlivé dáta z robota a jeho senzorov, následne posielať nejaké dáta, príkazy naspäť do robota a ku koncu samotné ladenie a ošetrovanie. K zjednodušeniu bude opäť pripravená kostra programu, ale bude podstatne ťažšie ju doplniť ako v prechádzajúcich návodoch.

## 3.4 Ľudský faktor

Pri návode je dôležitý lektor. Ten musí byť skúsený s programovaním ako pre Android tak pre Arduino, vopred tieto návody vyskúšal a prípadne si ich upravil podľa seba. A to z toho dôvodu, aby vedel zareagovať na akúkoľvek nečakanú situáciu, ktorá sa môže vyskytnúť.

### 3.5 Návrh aplikácie

Pri návrhu návodu je treba uvažovať aj tvorbu samotnej aplikácie. Pre tento účel nie je nutné, aby užívateľské rozhranie aplikácie bolo zamerané na dizajn, ale na funkčnosť. Je potrebné, aby každému senzoru robota bol priradený nejaký prvok, ktorý bude slúžiť ako výstup a v prípade pohybu (motorov) ako vstup. Užívateľské rozhranie musí obsahovať nasledovné prvky:

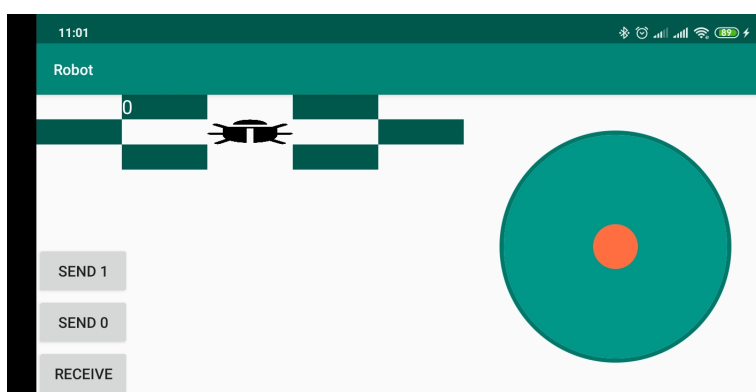
- Joystick, gyroskop v smartfóne alebo tlačítka na ovládanie pohybu.
- Textové výstupy na hodnoty zo senzorov vzdialenosti
- Textové výstupy na hodnoty z IMU
- Tlačítka na nastavovacie ako sú napríklad spárovanie Bluetooth zariadenia

Druhým prvkom pri návrhu je aj samotná funkcionálnosť. Pri programovaní funkcionality je dôležité povolenie aplikácie k prístupu k Bluetooth. Komunikácia medzi zariadeniami sa ustanovuje spárovaním smartfónu a Bluetooth modulu napojeného na Arduino. Po komunikácii nastáva prijímanie hodnôt, ich spracovanie a nakoniec prípadné posielanie príkazov naspäť do robota. Celý tento cyklus by sa mal pravidelne opakovať a to čo najrýchlejšie, avšak pokiaľ by bolo príliš rýchle, môžu nastať problémy s výkonom. Preto je vhodné zvoliť akýsi kompromis medzi týmito požiadavkami.

Pre ovládanie robota sa ponúkajú 3 možnosti:

- Joystick, ktorý sa spomína vyššie. Existuje množstvo voľne dostupných knižníc, ktoré by boli k tomuto projektu dostatočné
- Gyroskop, ktorý je priamo zabudovaný do operačného systému Android, ktorý tieto hodnoty neustále posiela pri akejkoľvek zmene.
- Tlačítka, ktoré by však neboli praktické, pretože by boli málo citlivé.

Referenčný návrh UI je nasledovný:



Obrázek 3.1: Návrh UI

Na tomto obrázku sú vidieť nasledovné prvky:

- Joystick na ovládanie pohybu a textový výstup hodnôt z joysticku

- Šesť textových výstupov okolo robota pre senzory vzdialenosti
- 3 Textové výstupy pre IMU

V budúcnosti môže nastať situácia, že použité knižnice nebudú už k dispozícii alebo ich bude potreba aktualizovať alebo návody sa budú rozširovať. Preto je potrebné, aby všetky návody poskytovali možnosť úpravy. Preto budú napísané v  $\text{\LaTeX}$ e a bude vyexportované nielen PDF ale aj zdrojové kódy. Taktiež budú napísané v dvoch jazykoch - Angličtina a čeština.

## Kapitola 4

# Referenčné riešenie

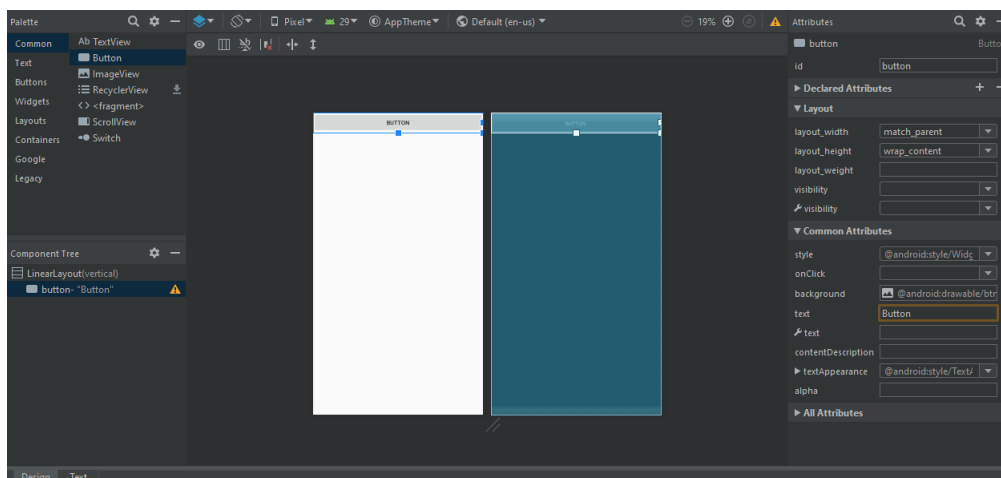
Táto časť obsahuje zostavenie programu, ktorý slúži aj ako referenčné riešenie toho projektu. Následne je riešenie upravené tak, že niektoré časti sú vymazané a zostáva na študentoch, aby ich doplnili. Vymazané časti sú vybrané podľa obtiažnosti. Pri zostavovaní programu je potrebné brať do úvahy aj zapojenie Bluetooth modulu k Arduino a samotné naprogramovanie Arduina.

### 4.1 Programovanie

Programovanie pre Android je postavené na jazyku Java. Základnou časťou je aktivita (angl. activity). Táto aktivita predstavuje okno aplikácie a jej kontext. Aktivít je možné vytvoriť viacej, avšak vždy je len jedna aktívna na popredí. Ostatné zostávajú bežať na pozadí a je možnosť sa k nim vrátiť. Štruktúra programu je podobná ako v iných programovacích jazykoch. Na začiatku aktivity sa vždy nachádzajú doložené knižnice. Knižnicu je možné doložiť pomocou kľúčového slova `import`, ale jednoduchšie je pomocou nariadenia pri samotnom písaní programu. Na chýbajúce knižnice Android Studio upozorní. Druhou časťou je samotné telo programu, ale na rozdiel od iných programovacích jazykov sa neoznačuje kľúčovým slovom `main`, ale nesie meno funkcie `onCreate`. To má ale presne tú istú funkciu ako `main`, teda zavolá sa hneď pri spustení danej príslušnej aktivity. Typicky sa tu nastaví layout aktivity, inicializujú premenné a prvky užívateľského rozhrania a volajú ďalšie definované funkcie. Ďalej Android poskytuje ďalšie funkcie ako sú `onResume`, `onBackPressed`, `onFinish` atď. Tieto funkcie sú využívané väčšinou pre správu aplikácie a správne ukončenie. Napríklad `onBackPressed` ukončuje aktuálnu aktivitu alebo program v prípade jedinej aktivity. V nej sa typicky volajú funkcie na správu pamäti a správne ukončenia danej aktivity. Tieto funkcie sú označené ako `@Override`. To znamená, že už sú vopred definované, ale pomocou preťaženia tejto metódy je možno doplniť prípadne zmeniť jej funkcionality.

Ďalšou časťou programovania je vytvorenie grafického dizajnu a rozloženie užívateľských prvkov, ako sú typicky tlačítka, textové polia, ovládacie prvky atď. To sa odohráva v súboroch `res`. Grafické rozhranie býva definované pomocou formátu XML, kde každý prvok predstavuje jeden element. Existuje však aj grafický dizajnér, kde systémom drag and drop je možné vytiahnuť prvky a umiestniť ich do pripraveného okna. Je to zdanlivo jednoduchšie, ale často býva problém s rozlíšením na viacerých zariadení a býva to nepresné, pretože sa tam napevno určujú súradnice prvkov. Ďalej je možné nastaviť rotáciu, rozlíšenie, API, tému a jazyk, ktoré môžu viesť napríklad k inému rozloženiu. Preto je to vhodné všetky tieto vlastnosti uvažovať do vytvárania dizajnu aplikácie. V súboroch `res` je možné ešte definovať

farby, reťazce znak a ich preklady (používajú sa najčastejšie pre výpisy na obrazovkách), ikony a obrázky.



Obrázek 4.1: Android Studio dizajnér

Ďalšími dôležitými časťami sú Gradle skripty, ktoré definujú preklad programu. Tu je možné priložiť neštandardné knižnice napríklad z githubu, meniť minimálnu cieľovú verziu Androidu a SDK, meniť internú verziu programu. Po akejkoľvek zmene je treba projekt zosynchronizovať, aby sa zmeny premietli.

Posledným dôležitým súborom je súbor Manifest. Ten sa vygeneruje sám, ale je potrebné ho meniť a upravovať pri vytvorení novej aktivity. Manifest teda obsahuje zoznam všetkých aktivít a je možné pri nich meniť aj nejaké základné údaje o konkrétnej aktivite, napríklad orientáciu. Manifest je opäť vo formáte XML.

## 4.2 Zapojenie Arduina k modulu a senzorum

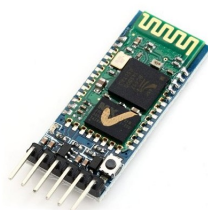
Z hľadiska k tomuto projektu sú k Arduinu pripojené 4 dôležité, ktoré je potrebné implementovať pre funkčnosť:

- Bluetooth modul zabezpečujúci komunikáciu
- Sensory vzdialenosti umiestnené po bokoch robota pre kontrolu vzdialenosti senzoru od najbližšej prekážky
- Jednotka IMU, ktorá určuje zmenu orientácie
- Motory s drivermi, ktoré zabezpečujú pohyb robota

### 4.2.1 Bluetooth HC-05

Bluetooth modul HC-05 [10] je komunikačný modul, ktorý dovoľuje pripojiť zariadenie (napr. Arduino) s iným zariadením s podporou Bluetooth. Obsahuje Bluetooth verziu 2.0 a pri komunikácii využíva sériovú linku s rýchlosťou 9600 baudov. Vzhľadom k malej anténe je dosah maximálne 10 metrov. Tento modul potrebuje napájanie v rozsahu 3,3 až 6 Voltov.

Prúdový odber pri 5 Voltoch je približne 2 miliAmér v klude, ale pri komunikácii až 40 miiliAmpér.



Obrázek 4.2: Blueooth modul HC-05

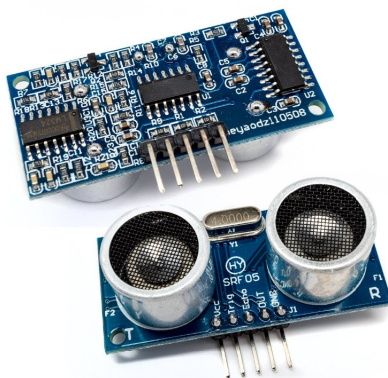
Použitý modul HC-05 obsahuje 6 pinov, z ktorých sú použité len 4:

- Vcc - vstupné napätie, ktoré pripojené na 5V Arduina
- Gnd - zem, ktorá je pripojená na GND Arduina
- Tx - vysielateľ, ktorý je pripojená na akýkoľvek digitálny pin Arduina
- Rx - prijímač, ktorý je pripojená na akýkoľvek digitálny pin Arduina
- Enable (nepoužitý)
- State (nepoužitý)

TX a RX je možné v prípade Arduina Mega napojiť na piny 10, 11, 12, 13, 14, 15, 50, 51, 52, 53, A8 (62), A9 (63), A10 (64), A11 (65), A12 (66), A13 (67), A14 (68), A15 (69).

#### 4.2.2 Sensory vzdialenosti

Ďalšie senzory, ktoré budú merať vzdialenosť sú ultrazvukové senzory [9], ktoré obsahujú merací senzor HY-SRF05. Pracuje na napájacom napätí 5 Voltov. Prúdový odber je približne od 2 miliAmpér po 40 miliAmpér. Pracujú na princípe vysielania a spätnej detekcie signálu na frekvencii 40kHz. Rozsah merania je približne od 2 cm až po 4,5 m.



Obrázek 4.3: Ultrazvukové senzory vzdialenosti

V tomto projekte je ultrazvukových senzorov vzdialenosti zapojených 6 po obvode. To znamená, že každý senzor zachytáva 60 stupňov - 2 sú v prednej časti, 2 po bokoch a 2 v zadnej časti robota. Cieľom tohoto senzoru je periodicky (napríklad každú sekundu) poselať vzdialenosť toho konkrétneho senzoru od najbližšej prekážky. Pri module HC-SR04 je zapojenie nasledovné:

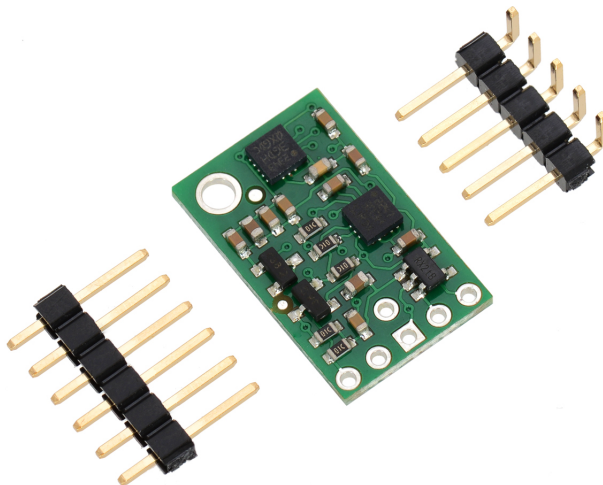
- Vcc - vstupné napätie napojené na 5V Arduina
- Trig - digitálny pin napojený na akýkoľvek pin Arduina
- Echo - digitálny pin napojený na akýkoľvek pin Arduina
- Gnd - zem, pripojená na Gnd Arduina

### 4.2.3 IMU

IMU [14] (z angl. Inertial measurement unit) je zariadenie, ktoré typicky zlučuje 3 ďalšie zariadenia:

- Gyroskop - zariadenie na meranie, alebo udržiavanie rovnakej orientácie, resp. rovnakého smeru.
- Akcelometer - zariadenie, ktoré umožňuje merať a analyzovať lineárne a uhlové zrýchlenie.
- Magnetometer - zariadenie na meranie veľkosti a smeru magnetickej indukcie alebo magnetického momentu (teda magnetizácie).

Pri tomto projekte je využitý konkrétne imu01c.



Obrázek 4.4: Zariadenie IMU

Tento senzor IMU je pripojený k Arduinu a vracia 3 údaje to vo formáte troch 16-bitových čísiel, ktoré predstavujú hodnoty gyroskopu, akcelometru a magnetometru. Tieto 3 údaje budú taktiež posielané periodicky a v aplikácii budú zobrazené.



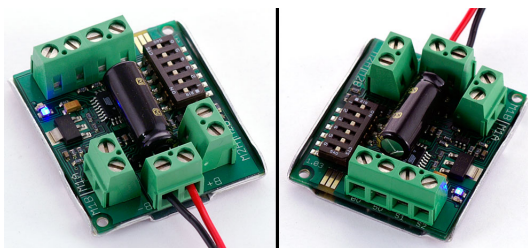
Zapojenie je pomocou zbernice I2C. I2C je synchronna zbernica, ktorá typicky obsahuje 2 vodiče - dátový a hodinový. Tento modul obsahuje 5 pinov a je zapojený nasledovne:

- Scl - hodinový signál, projený na SCL Arduina (pin 21)
- Sda - dátový signál, projený na SDA Arduina (pin 20)
- Gnd - zem, pripojená na GND Arduina
- Vcc - vstupné napätie, ktoré pripojené na 3,3V Arduina
- Vdd - napätie, nepoužité

Ďalej je nutné použiť 2 voľne dostupné knižnice - Wire.h a L3G.h, ktoré je možné stiahnuť priamo z vývojového prostredia Arduino IDE.

#### 4.2.4 Motor

Poslednou dôležitou súčasťou sú dva motory, pre ovládanie pohybu robota. Tie sú pripojené pomocou ovládača Sabertooth 2X5 [3]. Z každého motora vychádzajú 2 káble - pozitívny a negatívny, ktoré sú zapojené do M1A a M1B resp. M2A a M2B. Ďalej nasleduje zapojenie do Arduina:



Obrázek 4.5: Driver motora Sabertooth2X5

Driver Sabertooth2X5 môže pracovať až v šiestich módoch. V tom to projekte je využitý mód - Analog control, linear, independent.



Obrázek 4.6: Pohybový motor

Zapojenie je nasledovné:

- 0V - zapojené na GND Arduina
- 5V - zapojené na 5V Arduina
- S1 - zapojený do Arduina TX->1
- S2 - zapojený do Arduina RX->0

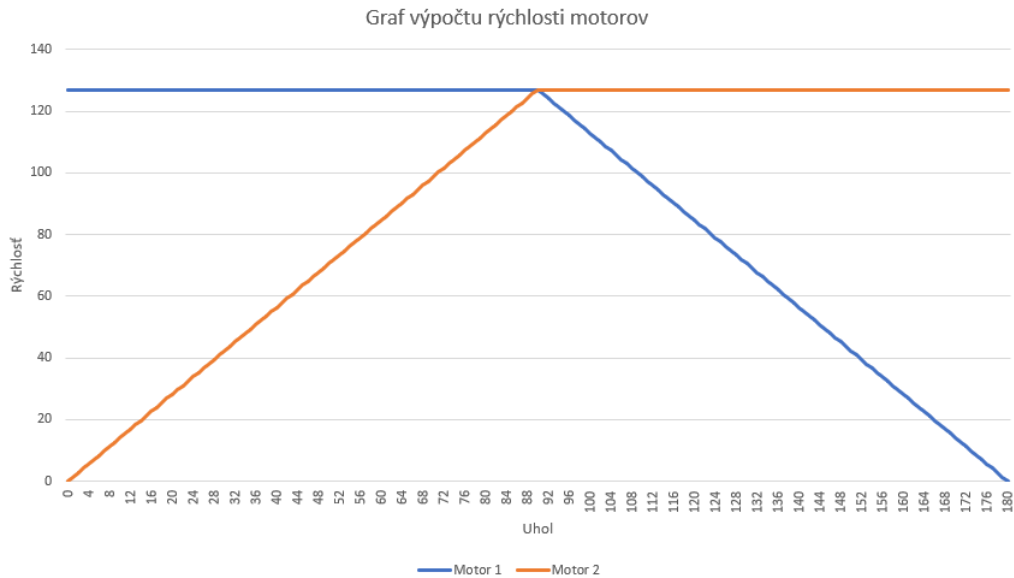
Pre tieto motory je možné využiť knižnicu SabertoothSimplified alebo Sabertooth. Pre knižnicu SabertoothSimplified je najpodstatnejšie využiť príkazy `ST.motor(x, y)`. X označuje číslo motoru 1 alebo 2. Y označuje smer, kde existujú hodnoty od -127 po 127. 127 je dopredu najrýchlejšie, 0 je zastavenie motora a -127 je dozadu najrýchlejšie.

Ako nastavenie maximálnej rýchlosti je použitý SeekBar (posuvná lišta), ktorý nadobúda hodnoty od 1 po 127. V závislosti na tom je prepočítavaná hodnota posiellaná motorom.

Ako výpočet pre motory možné využiť nasledovný vzorec: Pokiaľ uvažujeme, že maximálna rýchlosť je 127, tak existujú 4 základné smery, kde prvá hodnota udáva motor jeden a druhá hodnota udáva motor 2:

- Dopredu: 127 a 127
- Dozadu: -127 a -127
- Vpravo: 127 a 0
- Vľavo: 0 a 127

Vzhľadom k tomu, že robot môže mať problémy s pohybom dozadu, tak budeme uvažovať len tento jeden smer. Avšak dopredu môžeme uvažovať všetky smery od 0 až po 180 stupňov tak ako to zobrazuje nasledujúci graf. Zobrazuje rýchlosť motor v závislosti na stupni joysticku a v závislosti na maximálnej rýchlosti



Ako je vidieť z grafu, výpočet pre prvý motor v rozsahu od 0 po 90 je konštantný a to 127. Od 90 po 180 je však výpočet nasledujúci:

$$rychlost = (180 - uhol) * max/90$$

Pre druhý motor je výpočet konštantných 127 od uhlu 90 po 180. Od 0 po 90 je výpočet nasledujúci:

$$rychlost = uhol * max/90$$

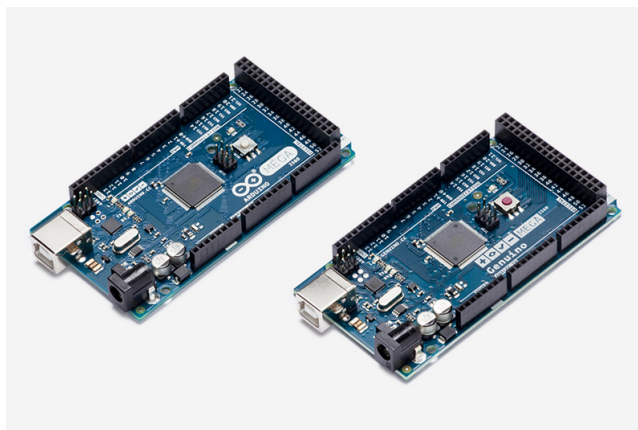
Je vhodné však uvažovať aj nepresnosť týkajúca pohybu prstu na joysticku. To znamená, že pokiaľ je uhol od 350 po 360 je možné to zaokrúhliť na 360 (resp. 0). To isté aj vľavo - pokiaľ je v rozmedzí 180 až 190, je možné uvažovať hodnotu 180. A ďalej je použité zaokrúhlenie pri pohybe dozadu. Tam je to väčšie, v rozmedzí od 240 po 300 sa uvažuje hodnota 270 - teda smer dozadu.

### 4.3 Program Arduina

Pre výber platformy je Arduino [1] jasná voľba. Existujú však aj alternatívy ako sú napríklad Parallax Basic Stamp, BX-24 od Netmedia, Phidgets, Handyboard od MIT. Avšak Arduino ich prevyšuje v týchto vlastnostiach:

- Cena sa pohybuje okolo 500 - 1000 Kč.
- Podporuje všetky operačné systémy - Windows, macOS a Linux
- Voľne dostupný softvér a programovacie prostredie

Pre konkrétny typ Arduina bola zvolená doska Arduino Mega 2560 [2], hlavne kvôli veľkému počtu pinov, kde by napríklad Arduino UNO nemuselo byť dostatočné. Arduino Mega 2560 je 8-bitová vývojová doska s 54 digitálnymi pinmi, 16 analógových vstupov a 4 sériové porty.



Obrázek 4.7: Arduino Mega 2560

Programovanie Arduina je založené na podmnožine jazyka C/C++ a využíva sa vývojové prostredie Arduino IDE (integrated development enviroment = integrované vývojové prostredie) a je voľne dostupné. Telo programu pozostáva z dvoch funkcií setup a loop. Setup slúži na inicializáciu typicky pinov. Loop je cyklus, ktorý sa neustále volá a opakuje.

Samotný program Arduina je upravený o jednoduchú funkcionálnosť. V nekonečnom cykle vykonávajú dve nasledujúce funkcie:

- Kontrola na príjem dát z Androidu cez Bluetooth modul. Pokiaľ sa na vstupe vyskytne nejaký príkaz a Arduino to rozpozna vykoná sa príslušná funkcia (napríklad pohyb dopredu).
- Posielanie dát z Arduina do Androidu cez Bluetooth modul zo senzorov ako sú senzory vzdialenosti, imu, atď. Tie si následne Android spracuje a vykoná príslušnú obslužnú rutinu (napríklad vypíše vzdialenosť, orientáciu, atď.)

## 4.4 Program Androidu

Po úspešnom zapojení a naprogramovaní Arduina prichádza čas na programovanie aplikácie pre Android. Hlavným problémom bolo vyriešiť samotnú komunikáciu medzi týmito odlišnými systémami. Preto existujú 2 aktivity, kde prvá slúži na ustanovenie spojenia a zároveň je to aj spúšťača aktivita - nazvaná **DevicesActivity**. Druhou aktivitou je samotné ovládanie a komunikácia medzi Androidom a Arduino - nazvaná **ControlActivity.java**. Dôležitou časťou programu je povolenie prístupu k Bluetooth, ktoré sa nastavuje v súbore Manifest.xml. To zahŕňa 2 povolenia:

- android.permission.BLUETOOTH
- android.permission.BLUETOOTH\_ADMIN

Ďalšou potrebnou časťou je pridanie knižnice pre joystick. To je možné realizovať v skripte Gradle.build pridaním riadku:

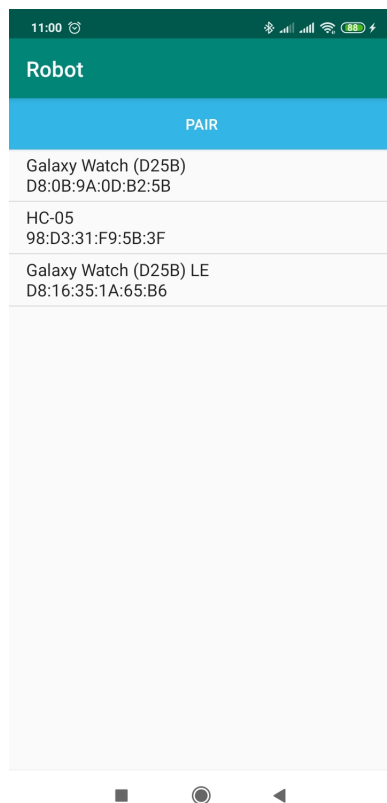
```
implementation 'io.github.controlwear:virtualjoystick:1.10.1'
```

prípadne inou verziou. Následne je potrebné program zosynchronizovať, aby sa aplikovali tieto zmeny.

V súbore manifest je pri druhej aktivite `ControlActivity.java` nastavená rotácia na šírku (angl. landscape), kvôli lepšiemu rozloženiu užívateľských prvkov

#### 4.4.1 DevicesActivity

Táto aktivita má len jednu úlohu - vyberá zariadenia, s ktorým bude operačný systém Android komunikovať zo zoznamu dostupných zariadení. Prvým krokom je vytvorenie spojenia. Predpokladom je však zapnuté Bluetooth v nastaveniach Androidu, pokiaľ však nie je zapnuté, aplikácia na to upozorní. Je potrebné na Androide vybrať zariadenie, s ktorým bude komunikovať, to znamená vybrať názov napríklad HC-05 a MAC adresu. Pomocou MAC adresy je možné rozlíšiť viacero zariadení. Najdôležitejšou časťou tejto aktivity je vyhľadanie dostupných zariadení. To je realizované pomocou zabudovanej Android Bluetooth knižnice, ktorá ponúka triedy typu `BluetoothAdapter`, ktorý slúži na získanie zariadení a `BluetoothDevice`, ktorý predstavuje samotné Bluetooth zariadenie. Ďalej táto aktivita obsahuje jednoduché prvky užívateľského rozhranie ako sú `ListView`, ktorý predstavuje zoznam zariadení, obsluhu tlačítka a spúšťanie novej aktivity. Pri spúšťaní novej aktivity je predávaná MAC adresa vybraného zariadenia.



Obrázek 4.8: DevicesActivity

#### 4.4.2 ControlActivity

Ďalšia aktivita predstavuje hlavnú časť celého projektu. Zahŕňa funkcie na naviazanie spojenia, prijímanie a odosielanie dát. K naviazaniu spojenia je potrebné poznať adresu druhého zariadenia, takže prvým krokom je získanie MAC adresy, ktorá bola predaná z predchádzajúcej aktivity. Keď už program pozná príjemcu, nasleduje vytvorenie komunikačného kanálu medzi programom a Bluetooth modulom (Arduinom). To zahŕňa vytvorenie Bluetooth schránky (socketu). To prebieha taktiež pomocou knižníc Androidu a tried typu BluetoothAdapter a BluetoothSocket. Ak vytvorenie schránok a pripojenie prebehlo v poriadku, môže nasledovať samotná výmena dát. Pre posielanie príkazov v smere z Androidu do Arduina je jednoducho vytvorený OutputStream, do ktorého sa zapíše konkrétny príkaz. V opačnom smere, teda z Arduina do Androidu je to trochu komplikovanejšie. Po úspešnom spojení sa spúšťa vlákno, ktoré v cykle a s určitým časovým intervalom spracováva dáta. Interval je možné jednoducho zmeniť, ale doporučená hodnota je 1 sekunda, teda 1000 milisekúnd. V tomto cykle sa spracuje jeden riadok z InputStreamu. Nasleduje zmena na užívateľskom rozhraní, ktorá však musí byť zaobalená v `runOnUiThread()`. Bez toho by sa zmena nepremietla na užívateľskom rozhraní, pretože sa pracuje v inom vlákne.

#### 4.4.3 Spracovanie dát

Spracovanie dát z Arduina zahŕňa spracovanie z týchto senzorov:

- Sensory vzdialenosti - Ten vracia jedno číslo, ktoré sa ľahko prepočítava na cm. Prepočítavanie je realizované buď v Androide alebo ešte predtým v Arduine.
- Senzor IMU - Tento senzor vracia 3 celé čísla, ktoré sa len zobrazujú na výstupe v aplikácii. Sú to hodnoty gyroskopu, akcelerometru a kompasu

#### 4.4.4 Užívateľské rozhranie

O užívateľskom rozhraní musí platiť, že má byť intuitívne, jednoduché a musí spĺňať všetky požiadavky. To zahŕňa tieto prvky:

- Tlačítka alebo joystick na ovládanie pohybu
- Výpis dát zo senzorov
- Nastavenie a spárovanie týchto dvoch systémov
- Ošetrenie chýb

Pre joystick je využitá voľne dostupná knižnica `virtualjoystick` [11], ktorá je pre tento projekt dostačujúca. Pri pohybe týmto prvkom vracia 2 hodnoty - uhol a silu. Uhol obsahuje hodnoty v stupňoch od 0 po 360, sila obsahuje hodnoty od 0 po 100. Namiesto sily je však použitý seekbar (posuvná lišta), ktorá je presnejšia na kontrolovanie a udržanie konštantnej rýchlosti.

Výpis dát zo senzorov vzdialenosti je realizovaný vo forme textových prvkov (v Androide `TextView` alebo `EditBox`). V tomto prípade sa tu nachádza 6 senzorov, ktoré sú rozložené po obvode robota, teda každý sníma 60 stupňov. Tomu je aj prispôsobené rozhranie, ktoré vypisuje vzdialenosť od prekážky a farebne to zafarbuje - ak senzor nasníma 10 cm, tak je to červené, ak je to v rozmedzí 10 - 20 cm, tak žlté, a ak je to viac ako 20 cm, tak je

to zelené. Ďalej obsahuje 3 textové polia pre výpis z IMU a 2 textové polia pre výpis z motorčeku

Prvým krokom však musí byť v operačnom systéme Android cez Bluetooth nájsť modul HC-05 a spárovať. Heslo pre spárovanie býva obvykle 1111 alebo 1234. Následne sa môže spustiť aplikácia. Je dôležité, aby mal program povolenie pre používanie Bluetooth (v súbore manifest) a aby bolo zapnuté (ak nie je, aplikácia na to upozorňuje). Po spustení programu nasleduje obrazovka, kde sa vypíšu všetky dostupné zariadenia pre spárovanie. Po kliknutí na modul HC-05 a úspešnom spojení prechádza na hlavnú obrazovku, kde prebieha spracovávanie údajov a pohyb joysticku, tieto časti sú už popísané vyššie.

# Kapitola 5

## Návod

Pri tvorbe návrhu je dôležité zaoberať sa nasledujúcimi vlastnosťami:

- Cielová skupina - návod musí byť prispôsobený skúsenostiam cieľovej skupiny, pre ktorú je tvorený. Nesmie byť príliš jednoduchý, ale ani je príliš zložitý.
- Jasnosť - To čo vysvetľuje musí byť jasné. Pokiaľ to nevyplýva priamo z tohoto návodu, musí to byť vysvetlené ústne.
- Následnosť - Postupuje od najjednoduchších vecí k zložitejším. V tomto prípade sa jedná napríklad o orientovanie vo vývojovom prostredí, programovanie v jazyku Java a tvorba užívateľského rozhrania a až potom samotná funkcionálnosť.
- Prehľadnosť - Návod musí byť prehľadný a doplnený obrázkami.

Návody pozostávajú z kódov, z ktorých sú vytrhnuté určité časti. Pri chýbajúcich častiach je vždy komentár, s postupom, ktorý má študent sledovať. Tieto časti komentárov začínajú vždy textom TODO:. Ako sa v predchádzajúcich kapitolách spomína, existujú 3 formy návodov rozdelené podľa cieľových skupín.

### 5.1 Začiatníci

U začiatníkov sa predpokladá, že nemajú žiadne alebo len minimálne skúsenosti s programovaním, preto je potrebné vysvetliť všetko, ale jednoducho a nie dopodrobna.

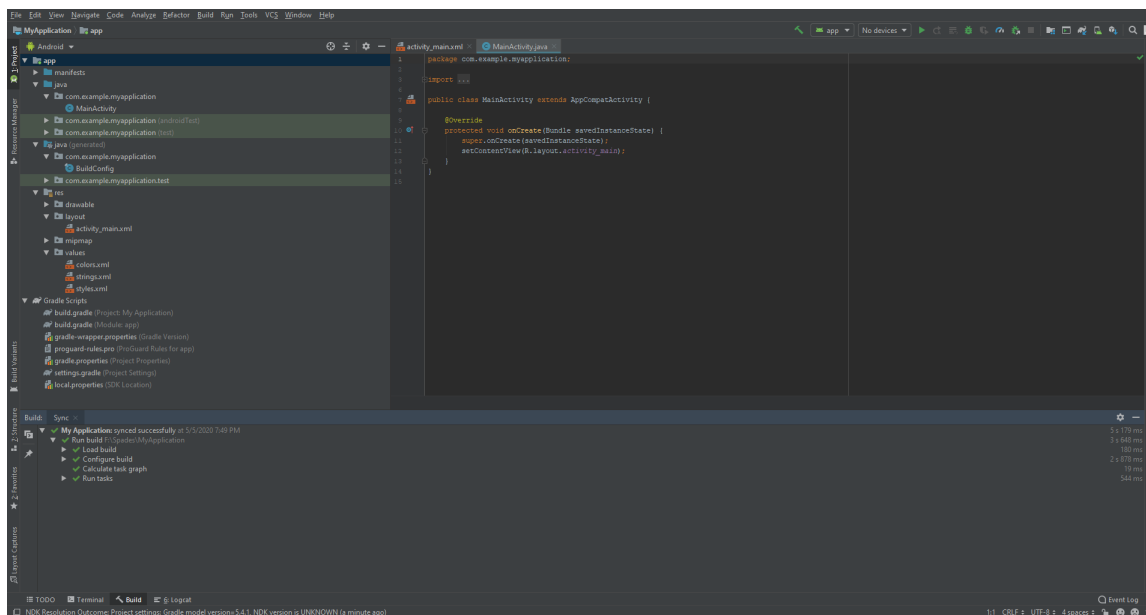
#### 5.1.1 Vývojové prostredie

Prvou kapitolou pre začiatníkov je zoznámenie sa s vývojovým prostredím. To obsahuje tieto kroky:

- Popis vývojového prostredia
- Projektová štruktúra programu
- Vytvorenie projektu, tzv. Hello World!
- Vysvetlenie ovládacích prvkov - preklad, spustenie, debug, atď.
- Vysvetlenie štruktúry projektu a prepojenie všetkých častí



Popis vývojového prostredia zahŕňa všetky dôležité ovládacie prvky, ktoré sú potrebné k samotnému programovaniu pre Android.

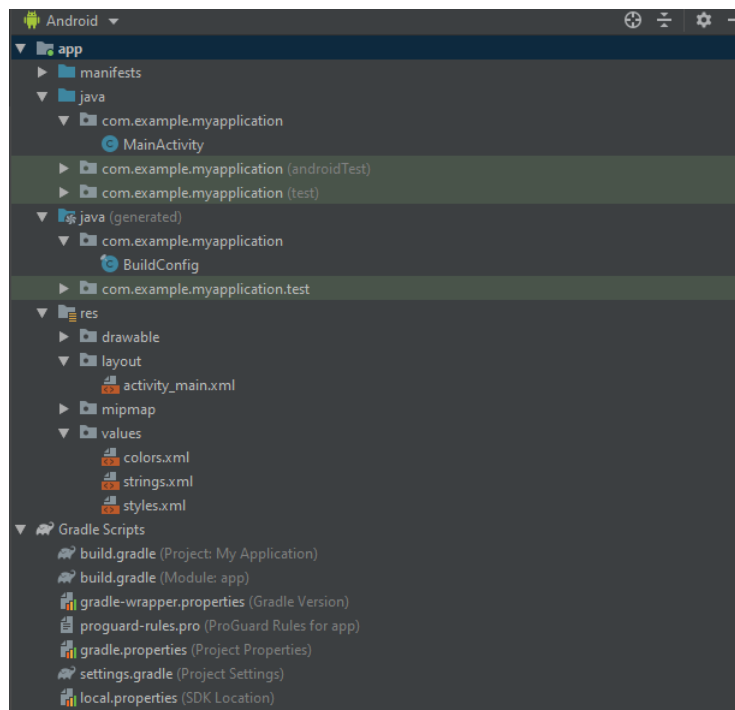


Obrázek 5.1: Vývojové prostredie

### 5.1.2 Projektová štruktúra

Projektová štruktúra programu vysvetľuje 4 základné typy súborov, ktoré programátor potrebuje poznať pri programovaní v Android Studiu. To zahŕňa:

- Gradle skripty, ktoré slúžia na zbuildovanie (preklad) projektu
- Manifest definuje aktivity
- Java obsahuje zdrojové Java alebo Kotlin súbory potrebné na beh a obsluhu aplikácie.
- Súbory res obsahujú všetky neprogramátorské súbory ako XML rozhrania, UI reťazce a obrázky.

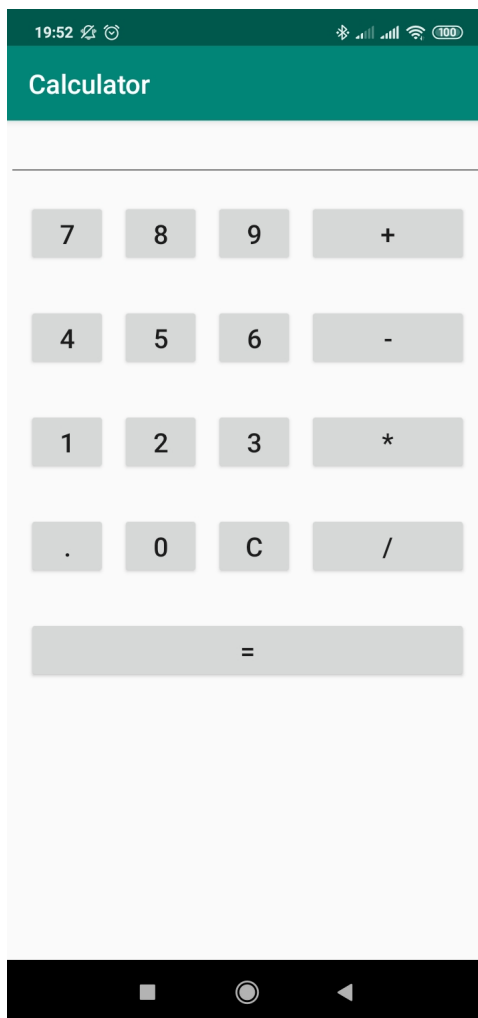


Obrázek 5.2: Štruktúra projektu

### 5.1.3 Tvorba jednoduchého programu

Keď už sú začiatočníci zoznámení vývojovým prostredím, je vhodné, aby si niečo vyskúšali naprogramovať sami. Preto je druhou časťou vytvorenie veľmi jednoduchého programu, ako je napríklad kalkulačka, kde sa zoznámia s tvorbou užívateľského rozhrania v dizajnéri, ale taktiež si skúsia naprogramovať jednoduchú funkcionálnosť. Pri tomto príklade kalkulačka, je vhodné implementovať len základné operácie ako sú súčet, rozdiel, podiel a násobenie.

V pri tvorbe užívateľského prostredia programu sa zoznámia so základnými prvkami ako sú textové polia, tlačítka.



Obrázek 5.3: Kalkulačka

Pri napájaní funkcionality na užívateľské prostredie sa zoznámia s konštrukciami, obsluhou tlačítiek a textových polí.

#### 5.1.4 Úvod do ovládania robota

Tretou časťou je spoznanie robota. Cieľom toho je zistiť, akým spôsobom pracuje robot. To obsahuje vysvetlenie nasledujúcich častí:

- Čo je to vlastne Arduino
- Vysvetlenie senzorov a ich zapojenie
- Vysvetlenie Bluetooth modulu a jeho zapojenie

#### Výmena dát

Pre začiatočníkov je náročné vytvárať nejakú komunikáciu. Preto sú tieto funkcie už pre nich naprogramované.

### 5.1.5 Spracovávanie dát a reprezentácia

Pri spracovaní dát je dôležité uvedomiť si význam jednotlivých dát, spracovať formát prijatých dát a následne ich interpretovať. Formát prijatých dát by mal byť jednoduchý - zvolené je teda výpis jednotlivých hodnôt, ktoré sú oddelené bodkočiarkou. Je to teda: [distance1];[distance2];[distance3];[distance4];[distance5];[distance5];[imu1];[imu2];[imu2] kde distance1 až distance5 sú senzory vzdialenosti, imu1, imu2 a imu3 je hodnota gyroskopu, akcelerometru a magnetometru.

Po úspešnom spracovaní toho formátu dát nasleduje reprezentácia, kde je dôležité zaistiť, aby sa každá hodnota vypísala konkrétnemu prvku užívateľského rozhrania.

### 5.1.6 Záver

Poslednou časťou je záver, kde sa overí funkčnosť všetkých častí. Je potrebné, aby študenti využili funkciu debugu, aby si overili správnosť všetkých hodnôt a aby sa s debugom zoznámili viacej.

## 5.2 Pokročilí

U pokročilých sa zase predpokladá, že už vedia niečo programovať, aj keď to nemusí byť úplne programovanie v Androide. To znamená, že je potrebné vysvetliť podobné veci ako pri začiatkoch, ale trochu zložitejšie a nie tak dopodrobna.

### 5.2.1 Vývojové prostredie a program

Opäť je potrebné začať spoznaním užívateľského prostredia, ale rovno to spojiť s nejakým jednoduchým programom ako u začiatkov. Tu je ale vhodné využiť zložitejšie konštrukcie - užívateľské prostredie napísať v XML, a pri funkcionalitu implementovať zložitejšie operácie.

### 5.2.2 Úvod do ovládania robota

Táto kapitola je podobná ako pre začiatkov. Je teda potrebné vysvetliť princíp Arduino robota, vysvetliť senzory a komunikačný Bluetooth modul. Na rozdiel od začiatkov je však vhodné ukázať zapojenia jednotlivých komponent a vysvetliť podrobnejšie. Taktiež je vhodné ukázať program Arduina (zjednodušene), aby študenti vedeli na akom princípe pracuje a vedeli ho upraviť podľa seba

### 5.2.3 Vytvorenie komunikácie s robotom

Cieľom pokročilých študentov v tejto úlohe je doplniť chýbajúce výrazy vo funkciách tak, aby aj pochopili princípy a význam jednotlivých konštrukcií.

### Komunikácia z Androidu do Arduina

Na strane Androidu je potrebné vytvoriť komunikačný kanál pomocou Bluetooth socket, kde je potrebné naviazať spojenie pripraviť funkcie na posielanie dát. Následne túto funkcionalitu volať.

Na strane Arduina je potrebné v nekonečnom cykle doplniť časť funkcionality, ktorá bude spracovávať jednotlivé príkazy z Androidu. Príkladom je príkaz na pohyb - po obdržaní povelu, je nutné zavolať správnu funkciu so správnymi parametrami

### **Komunikácia z Arduina do Androidu**

Na strane Androidu je potrebné doplniť komunikačný kanál o spracovanie prijatých dát a následne doplniť do volacieho vlákna (angl. thread).

Na strane Arduina je potrebné doplniť jednoduchú funkciu v cykle na posielanie dát, ktoré sú získavané zo všetkých senzorov. Formát týchto dát je zvolený študentom.

#### **5.2.4 Spracovávanie dát**

Hlavným rozdielom v tejto kapitole je množstvo kódu, ktoré aby študenti doplnili. Princíp zostáva rovnaký. Ďalším rozdielom je formát dát - ten študent sám zostavuje.

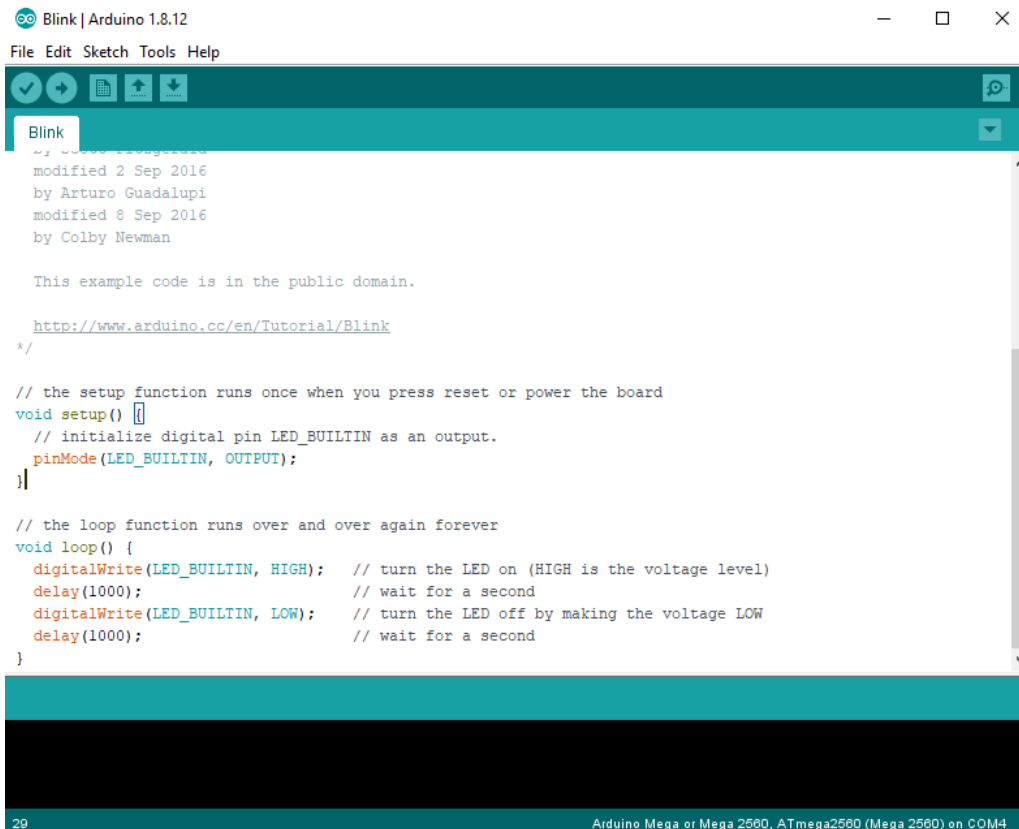
#### **5.2.5 Záver**

Poslednou časťou je opäť záver, kde sa overí funkčnosť všetkých častí. Znova si môžu overiť funkciu debugeru. Pokiaľ je všetko funkčné, môžu pokračovať na práci na užívateľskom rozhraní - to obsahuje hlavne grafický dizajn a rozloženie prvkov.

### **5.3 Skúsení**

Pre skúsených nemá význam vysvetľovať užívateľské prostredie, pretože Android Studio je veľmi jednoduché a podobá sa klasickým vývojovým prostrediam ako sú CodeBlocks, Eclipse, NetBeans, atď. Preto stačí si u nich otvoriť len kostru programu a vysvetliť štruktúru projektu.

Ďalšou časťou je vysvetlenie programovania v vývojovom prostredí Arduino IDE. To je však ešte jednoduchšie ako Android Studio.



Obrázek 5.4: Arduino IDE

### 5.3.1 Robot

Prvou časťou je zoznámenie s robotom, to zahŕňa:

- Arduino Mega
- Senzory a ich zapojenie
- Arduino HC-05 Bluetooth modul a jeho zapojenie
- Arduino kód - správa senzorov a tvorba komunikácie s Androidom

### 5.3.2 Úloha vytvorenie užívateľského rozhrania

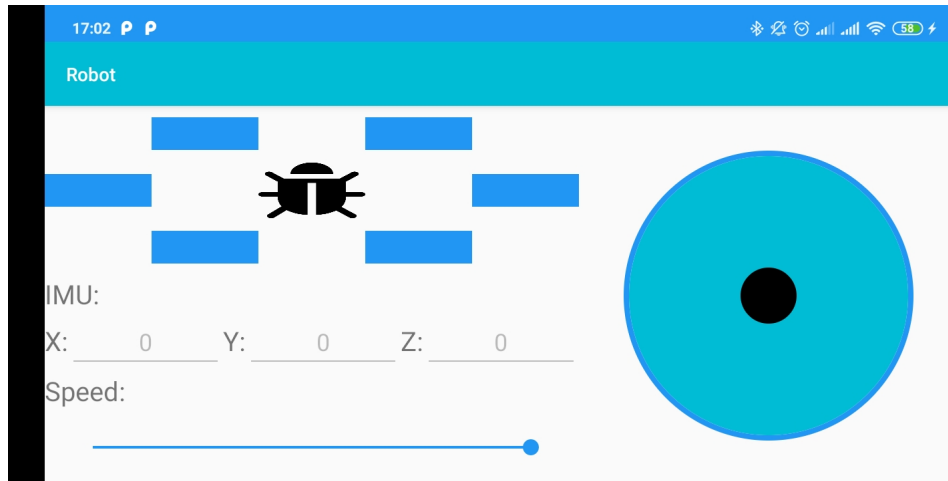
Pri tejto úlohe je potrebné vytvoriť užívateľské rozhranie tak, aby odpovedalo referenčnému riešeniu. Pre skúsených nemá zmysel vysvetľovať význam jednotlivých prvkov, pretože sa s tým už stretli. Študenti majú k dispozícii obrázky referenčného riešenia a ich cieľom je vytvoriť také riešenie, aby čo najviac odpovedalo referenčnému. Do úvahy je však potrebné brať nielen rozmiestnenie, ale rôzne rozlíšenia. Preto musia dbať na to, aby bolo ich riešenie nielen správne, ale aj univerzálne.

To znamená vytvoriť a rozmiestniť nasledovné prvky:

- 6 textových polí pre hodnoty zo senzorov vzdialenosti - k tomu je možné využiť EditText alebo TextView

- 3 textové polia pre hodnoty z IMU podobne ako vyššie
- Joystick pre ovládanie smeru motorov - JoystickView
- Posuvnú lištu (seekbar) na ovládanie rýchlosti motorov

Ďalej je možné využiť konštrukcie ako sú Space, LinearLayout a taktiež šírku a výšku prvkov upravovať relatívne pomocou atribútu layout\_weight.



Obrázek 5.5: Uživatelské rozhranie

### 5.3.3 Úloha spárovanie zariadení

Pri tejto úlohe je potrebné doplniť funkcionalitu v aktivite DevicesActivity ktorá bude zabezpečovať vyhľadanie a zvolenie zariadenia z dostupných Bluetooth zariadení. Cieľom je teda napojenie už pripraveného užívateľského rozhrania na funkcionalitu. Okrem typických Android konštrukcií je potrebné využiť triedy BluetoothAdapter a BluetoothDevice spomínané predošlých kapitolách. Riešenie tejto úlohy pozostáva teda z 2 krokov:

1. Po spustení zistiť stav Bluetooth Androidu. Pokiaľ je vypnuté, upozorní na to a nepokračuj ďalej. Pokiaľ je zapnuté pokračuj na ďalší krok.
2. Získaj si dostupné Bluetooth zariadenia a ulož ich do pripraveného poľa zariadení. Toto pole sa ďalej zobrazí a prepíše po stlačení tlačítka na obnovenie.

### 5.3.4 Vytvorenie komunikácie

Cieľom tejto kapitoly, je vytvorenie základnej komunikácie. To znamená, posielanie dát smerom z Androidu do Arduina a opačne. Študenti majú pripravenú kostru programu, ktorú je nutné doplniť. Ďalšou časťou je navrhnuť protokol alebo syntax komunikácie, ktorý bude pre robota a Android program jednotný. Pre zjednodušenie a pre minimalizáciu prenesených dát je vhodné využiť len jednoduché výpisy hodnôt oddelené nejakým oddeľovačom. Formáty typu XML alebo JSON by boli síce komplexnejšie, ale nie je to potrebné.

## Komunikácia z Androidu do Arduina

Na strane Androidu je potrebné vytvoriť asynchrónnu úlohu (angl. AsyncTask), ktorá zabezpečuje obojstrannú komunikáciu a následne využívať jej funkcie. Najpodstatnejšou súčasťou je vytvorenie Bluetooth schránky (angl. socket). Následne pomocou tejto schránky poslať dáta pomocou konštrukcie OutputStream.

Na strane Arduina je potrebné v cykle loop() využiť funkciu bluetooth.read(), ktorý v bytoch (typ byte), získava prijaté dáta. Predtým je nutná inicializácia a vhodné je kontrolovať dáta pomocou bluetooth.available().

## Komunikácia z Arduina do Androidu

Na strane Androidu sa opäť využíva asynchrónna úloha ako pri opačnej komunikácii (viz. vyššie). Následne je však potrebné vytvoriť vlákno (angl. Thread), ktoré pracuje na pozadí a v nekonečnom cykle periodicky (napríklad každých 60 sekúnd) vykonáva funkciu, kde sa získavajú z Bluetooth schránky (angl. socketu) vstupné dáta z konštrukcie InputStream. Tie sa ďalej spracovávajú pomocou BufferedReader. Potom je z neho možné získať jeden riadok a ukončiť. V tom jednom riadku sa vyskytujú všetky hodnoty zo senzorov, ktoré Arduino posiela.

Ďalším problémom je zobrazenie dát v samotnom užívateľskom rozhraní v Androide. Tento problém je vyriešený pomocou funkcie runOnUiThread, bez ktorej by sa výsledky neprekresľovali alebo neprepisovali v užívateľskom prostredí aplikácie.

Na strane Arduina je potrebné opäť v cykle loop() využiť funkciu bluetooth.println(x), ktorá posiela dáta. Tieto hodnoty posiela v jednom riadku, preto je predtým potrebné ich poskladať a rozdeliť do zvoleného formátu.

### 5.3.5 Spracovávanie dát a reprezentácia

Po úspešnom získaní a posielaní akýchkoľvek dát prichádza na radu samotné spracovávanie a reprezentovanie dát. To obsahuje operácie, ktorých úlohou je z dát vybrať dôležité hodnoty a následne ich previesť a reprezentovať na užívateľskom rozhraní. Napríklad hodnoty zo senzorov vzdialenosti je ich treba prepočítať na jednotky cm a vypísať do textového poľa. Pokiaľ je to jednoduchý formát, teda čísla oddelené nejakým oddeľovačom, tak je to veľmi jednoduché. Dôležité je potom si dávať pozor len na správne spárovanie hodnôt. Ďalšou možnosťou je po spracovaní hodnôt poslať Arduino nejaký príkaz. Príkladom je opäť senzor vzdialenosti ale spolu s motorom. Pokiaľ sa zo senzoru vráti hodnota menšia ako napríklad 10 centimetrov, je nutné motorčeky zastaviť, aby nedošlo k nárazu.

### 5.3.6 Záver

Poslednou časťou je opäť záver, kde sa overí funkčnosť všetkých častí. Študenti musia ladit riešenie a kontrolovať správnosť všetkých naprogramovaných častí. Pokiaľ všetko funguje, je potrebné, aby boli získané hodnoty správne spárované s nimi priradenými textovými poľami. Ak to majú správne riešenie, prichádza na rad vylepšenie. To zahŕňa vylepšenie dizajnu, ovládania a funkcionality. Pri dizajne môžu upravovať grafické prvky, rozmiestnenie. Pri ovládaní môžu vyskúšať gyroskop zabudovaný v smartfóne. Ďalej je možnosť testovania hodnôt. Je možné upravovať interval, ktorým sa pravidelne posielajú a prijímajú dáta medzi Androidom a Arduino. Je vhodné experimentovať a navrhovať vhodnú hodnotu. V



závislosti na tom je treba upraviť aj minimálnu hodnotu, ktorá určuje pri akej hraničnej hodnote má poslať príkaz na zastavenie, pokiaľ sa objaví na senzore prekážka.

# Kapitola 6

## Testovanie

Testovanie bolo ovplyvnené malým počtom testovaných ľudí a to z dôvodu zavretia všetkých škôl. Tým pádom toto testovanie nemusí byť presné. Na testovanie boli vybraní ľudia, ktorí spadajú do danej kategórie. Testovanie prebiehalo tak, že testovým osobám bol vysvetlený návod presne takým štýlom, akým by to bolo vysvetlené študentom, aby boli výsledky a postrehy čo najpresnejšie.

### 6.1 Začiatovníci

Prvý krok bol zameraný na vývojové prostredie. To zahŕňalo orientácia v ňom, vysvetlenie projektovej štruktúra, vytvorenie projektu, spustenia, prekladu a ladenia. Tu sa nevyskytli žiadne komplikácie. Pokiaľ však nastanú pochybnosti u študenta, je ľahké niečo vysvetliť, pretože to spolu súvisí.

Ďalším krokom bolo vytvorenie vzorového programu - kalkulačky. V tomto kroku sa študenti často strácali, pretože to bol ich prvý projekt. Je preto nutné, aby vyučujúci pracoval na tomto projekte s nimi. Rozdelil si to na časti a všetci postupovali spolu.

V ďalšej časti bolo vysvetlené Arduino a senzory. To študenti pochopili, pretože to bola len "teória".

V poslednej časti, kde programovali je potrebné sa zamerať a znovu pracovať súbežne so študentami. Pri užívateľskom rozhraní je potrebné vysvetliť dizajnéra (predpokladá sa, že sa použije ten, nie XML formát). V tejto úlohe sa často vyskytovali otázky aj z predchádzajúcich častí.

### 6.2 Pokročilí

Pri vývojovom prostredí títo študenti ľahšie chápali vysvetlenie, niektorí sa už s podobným stretli. Bolo však aj tak potrebné to vysvetliť podobne ako pri začiatovníkoch.

Pri vytváraní programu kalkulačka boli študenti samostatnejší a preto nebolo treba presne postupovať spolu. Bolo však treba vysvetliť a ukázať im niektoré programovacie funkcie.

Pri vysvetlení Arduina to bolo podobné ako pri začiatovníkoch.

V poslednej časti kde sa programovali, boli vidieť rozdiel medzi touto skupinou a začiatovníčkou. Skúsenosti s programovaním a s užívateľskými rozhraniami hrali dôležitú úlohu.

## 6.3 Skúsení

Skúsení študenti nemali žiadne problémy pri orientovaní sa vo vývojovom prostredí. Jediné čo bolo potrebné vysvetliť je projektová štruktúra.

Pri vysvetľovaní o Arduine nebol žiadny problém. Pravdepodobne by sa dalo zakomponovať aj zapojenie tohoto robota alebo niektorých častí na študentoch.

Pri programovaní boli opäť vidieť skúsenosti, ktoré hrali veľmi dôležitú úlohu.

## 6.4 Zhodnotenie

Po testovaní boli upravené návody nasledujúce:

- Väčšie obrázky v prezentáciách
- Odstránenie prebytočného textu v prezentáciách. Je to na prednášajúcom, aby tieto časti dovysvetľoval.
- Úpravy niektorých programovacích častí, hlavne zjednodušenie pre začiatočníkov.

Ďalšie zmeny, ktoré by sa dali zakomponovať:

- Zapojenie niektorých častí robota študentami.
- Rozšíriteľnosť ďalšími senzormi

# Literatura

- [1] *Arduino - Home*. 2020. Dostupné z: <https://www.arduino.cc/>.
- [2] *ARDUINO MEGA 2560 REV3*. 2020. Dostupné z: <https://store.arduino.cc/arduino-mega-2560-rev3>.
- [3] *Sabertooth dual 5A motor driver*. 2020. Dostupné z: <https://www.dimensionengineering.com/products/sabertooth2x5>.
- [4] ANDROIDDEVELOPERS. *Android Studio*. 2020. Dostupné z: <https://developer.android.com/studio>.
- [5] ANDROIDDEVELOPERS. *Android Studio tutoriál*. 2020. Dostupné z: <https://developer.android.com/guide>.
- [6] BUTTERFIELD, B. *Android Studio For Beginners*. 2017. Dostupné z: <https://www.youtube.com/watch?v=dF1PARW5IX8>.
- [7] FRIESEN, J. *Android Studio for beginners*. 2020. Dostupné z: <https://www.javaworld.com/article/3095406/android-studio-for-beginners-part-1-installation-and-setup.html>.
- [8] KANDI, J. *Learn how to develop for Android*. 2013. Dostupné z: <https://www.slideshare.net/apps4ucoza/meetup-learn-beyond-the-hello-world-android-app>.
- [9] M, L. *Měřič vzdálenosti ultrazvukový*. 2016. Dostupné z: <https://navody.arduino-shop.cz/navody-k-produktum/meric-vzdalenosti-ultrazvukovy.html>.
- [10] M., L. *Arduino Bluetooth modul HC-05*. 2017. Dostupné z: <https://navody.arduino-shop.cz/navody-k-produktum/arduino-bluetooth-modul-hc-05.html>.
- [11] MAKOWILDCAT. *Virtual-joystick-android*. 2018. Dostupné z: <https://github.com/controlwear/virtual-joystick-android>.
- [12] MALONE, D. *Android Studio Tutorial*. 2019. Dostupné z: <https://www.youtube.com/watch?v=g9YblXBQ5uU&>.
- [13] MULLIS, A. *Android Studio tutorial for beginners*. 2017. Dostupné z: <https://www.androidauthority.com/android-studio-tutorial-beginners-637572/>.
- [14] POLOLU. *MinIMU-9 v3 Gyro, Accelerometer, and Compass*. 2016. Dostupné z: <https://www.pololu.com/product/2468>.
- [15] SRIJAN, S. *Introduction to Android and Android Studio*. 2015. Dostupné z: <https://www.slideshare.net/SuyashSrijan/introduction-to-android-and-android-studio>.

- [16] TUTORIALSPPOINT. *Android Studio tutorial*. 2020. Dostupné z:  
[https://www.tutorialspoint.com/android/android\\_studio.htm](https://www.tutorialspoint.com/android/android_studio.htm).

# Příloha A

## Návody

Obsahom tejto prílohy sú tri návody, každý v češtine a aj v angličtine. Sú vo formáte  $\text{\LaTeX}$  spolu s použitými obrázkami a to z toho dôvodu, aby mohli byť tieto návody do budúcnosti upravované. Sú však ale aj vyexportované vo formáte PDF. Súčasťou návodov je aj program, bude použitý pri prezentácii týchto návodov a bude slúžiť pre študentov, aby do neho dopĺňali svoju funkcionality. Tento program je napísaný pre Android Studio. Všetko je to komprimované v jednom archíve, ktorý má nasledujúcu štruktúru:

- zaciatočníci - Zdrojové súbory k prezentácii a k projektu pre zaciatočníkov
- pokročili - Zdrojové súbory k prezentácii a k projektu pre pokročilých
- skuseni - Zdrojové súbory k prezentácii a k projektu pre skúsených

Súčasťou archívu zaciatočníkov a pokročilých je aj referenčné riešenie kalkulačka, ktorá slúži pre lepšie chápanie programovania v Android Studiu.

## Příloha B

# Program Android

Obsahom tejto prílohy je program, ktorý slúži ako referenčné riešenie. Program je funkčný. Je napísaný v Android Studiu. Program je komprimovaný do jedného archívu. Po rozbalení je tento program funkčný a preložiteľný vo vývojovom prostredí Android Studio. Po otvorení tohoto programu v Android Studiu to vyhodí hlášku po zmenu SDK.

Potrebné prerekvizity sú:

- Java development kit (JDK) verzia 5 a vyššie,
- Java Runtime Environment (JRE) 6
- Android Studio

## Příloha C

# Vzorový program Arduino

Tento program pre Arduino slúži ako postup riešenia, ktoré je potrebné pridať do konkrétneho riešenia projektu robota. Každý robot môže byť naprogramovaný inak, preto je nutné, aby bol obohatený o túto funkcionality. Slúži to však len ako akési referenčné riešenie, podľa ktorého je potrebné upraviť už existujúci program. Obsahuje to jeden zdrojový súbor `program.ino`, ktorý je preložiteľný vo vývojovom prostredí Arduino.