

Ovládání nakloněné roviny pomocí chytrého telefonu

Stanislav Moško

Bakalářská práce
2024

 Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Stanislav Moško**
Osobní číslo: **A20014**
Studijní program: **B0613A140020 Softwarové inženýrství**
Forma studia: **Prezenční**
Téma práce: **Ovládání nakloněné roviny pomocí chytrého telefonu**
Téma práce anglicky: **Control of Inclined Plane Using a Smartphone**

Zásady pro vypracování

1. Proveďte stručnou literární rešerši zaměřenou na embedded systémy pro ovládání náklonu.
2. Popište způsoby programování aplikací pro platformy iOS a Android. Popište práci se senzorickým systémem chytrého telefonu.
3. Vyberte komerčně dostupnou manuálně ovládanou nakloněnou rovinu a vhodné akční členy a řídicí jednotku pro její přestavbu na automatizované ovládání náklonu ve dvou osách. Případně vhodný systém vytvořte podle vlastního návrhu.
4. Sestavte a naprogramujte zařízení pro ovládání náklonu roviny. Řídicí jednotka bude schopna přijímat žádané hodnoty náklonu z nadřazeného systému.
5. Zvolte vhodnou mobilní platformu, která bude sloužit jako nadřazený systém ovládání náklonu roviny. Vytvořte aplikaci pro chytrý telefon, která bude na základě náklonu telefonu posílat do řídicí jednotky náklonu roviny žádané hodnoty náklonu.
6. Otestujte a odlaďte celý systém ovládání náklonu roviny pomocí chytrého telefonu a zhodnoťte jeho vlastnosti.

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

Seznam doporučené literatury:

1. IENNE, Paolo a Rainer LEUPERS. *Customizable embedded processors: design technologies and applications*. San Francisco, Calif.: Elsevier/Morgan Kaufmann, 2007, 1 online zdroj (xxviii, 497 p.). The Morgan Kaufmann series in systems on silicon. ISBN 9780080490984. Dostupné také z: https://proxy.k.utb.cz/login?url=http://app.knovel.com/hotlink/toc/id:kpCEPDTA06/customizable_embedded_processors__design_technologies_and_applications
2. ZHANG, Peng. *Advanced industrial control technology*. Amsterdam: William Andrew/Elsevier, 2010, 1 online zdroj (xxi, 842 pages). ISBN 9781437778076. Dostupné také z: https://proxy.k.utb.cz/login?url=http://app.knovel.com/hotlink/toc/id:kpAICT0003/advanced_industrial_control_technology
3. WARREN, John-David, Josh S. ADAMS a Harald MOLLE. *Arduino robotics*. [New York]: Apress, [2011], xxiv, 601 s. Technology in action. ISBN 9781430231837.
4. ALLEN, Grant. *Android 4: průvodce programováním mobilních aplikací*. Brno: Computer Press, 2013, 656 s. ISBN 9788025137826.
5. LACKO, Ľuboslav. *Vývoj aplikací pro iOS*. Brno: Computer Press, 2018, 479 s. ISBN 9788025149423.

Vedoucí bakalářské práce: **Ing. Petr Chalupa, Ph.D.**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **5. listopadu 2023**
Termín odevzdání bakalářské práce: **13. května 2024**



doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 5. ledna 2024

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor;
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

.....
podpis studenta

ABSTRAKT

Táto bakalárska práca sa zaoberá vývojom systému na diaľkové ovládanie naklonenej roviny pomocou mobilného telefónu. Systém využíva senzory mobilného zariadenia na spracovanie údajov o naklonení v reálnom čase. Práca kombinuje teoretické aspekty embedded systémov s praktickým vývojom aplikácie pre Android, ktorá komunikuje so serverom na Raspberry Pi. Vývoj aplikácie ukazuje, ako môžu moderné technológie zlepšovať interakciu medzi človekom a strojom prostredníctvom intuitívnych užívateľských rozhraní a pokročilého senzorického spracovania. Cieľom práce je demonštrovať, že použitie embedded systémov a mobilných technológií môže efektívne riešiť reálne technologické problémy, čím otvára nové možnosti pre automatizáciu a diaľkové ovládanie.

Kľúčová slova: Raspberry Pi, servomotory, Android, Internet, senzory

ABSTRACT

This bachelor's thesis explores the development of a system for remote control of an inclined plane using a mobile phone. The system utilizes sensors within the mobile device to process real-time data on inclination. The work combines theoretical aspects of embedded systems with the practical development of an Android application that communicates with a server on Raspberry Pi. The development of the application demonstrates how modern technologies can enhance human-machine interaction through intuitive user interfaces and advanced sensor processing. The aim of the thesis is to show that the use of embedded systems and mobile technologies can effectively solve real technological problems, thereby opening new possibilities for automation and remote control.

Keywords: Raspberry Pi, servomotors, Android, Internet, sensors

Chcel by som vyjadriť svoje úprimné poďakovanie všetkým, ktorí mi pomohli pri vypracovaní tejto bakalárskej práce. V prvom rade by som sa rád poďakoval vedúcemu práce, Ing. Petr Chalupa, Ph.D. za jeho ochotu, cenné usmernenie a trpezlivosť počas celého procesu tvorby práce.

Taktiež by som sa chcel poďakovať svojej rodine a priateľom za ich neustálu podporu, povzbudenie a pochopenie, bez ktorých by táto práca nebola možná. Osobitne by som sa chcel poďakovať mojej úžasnej mame, ktorá mi vytvorila ideálne podmienky na vypracovanie tejto práce a taktiež počas celého štúdia.

PodĎakovať by som sa tiež chcel spoločnosti TSS Group a.s. za ich ústretovosť pri plnení mojich študijných povinností.

Na záver by som chcel citovať motto, ktoré ma sprevádzalo počas celého môjho štúdia a ktoré odráža môj prístup k prekonávaniu výziev: „Who’s gonna carry the boats?“ Tento výrok ma motivoval prekonať všetky prekážky a dosiahnuť všetky ciele, ktoré som si stanovil.

OBSAH

ABSTRAKT	5
ABSTRACT	5
ÚVOD	9
I TEORETICKÁ ČÁST	10
1 EMBEDDED SYSTÉMY	11
1.1 AKO FUNGUJÚ EMBEDDED SYSTÉMY.....	11
1.2 TYPY EMBEDDED SYSTÉMOV.....	11
1.2.1 Samostatné embedded systémy.....	11
1.2.2 Real-Time Embedded Systémy.....	12
1.2.3 Sieťové embedded systémy.....	12
1.2.4 Mobilné embedded systémy.....	12
2 RASPBERRY PI	13
2.1 MODELY RASPBERRY PI.....	14
2.2 RASPBERRY Pi 5.....	14
2.3 ARCHITEKTÚRA ARM.....	15
2.3.1 BCM2712.....	15
2.3.2 Konektivita čipu BCM2712.....	16
2.4 GPIO.....	16
2.4.1 I2C.....	17
2.4.2 SPI.....	18
2.4.3 UART.....	18
2.4.4 GND.....	18
2.4.5 5V.....	19
2.4.6 3V3.....	19
2.5 WIFI.....	19
2.5.1 802.11b.....	19
2.5.2 802.11g.....	20
2.5.3 802.11n.....	20
2.5.4 802.11ac.....	20
2.5.5 2,4 GHz.....	20
2.5.6 5 GHz.....	21
2.6 BLUETOOTH.....	21
2.6.1 BLE.....	21
3 LINUX	23
3.1 RASPBIAN.....	23
3.1.1 Debian 12.....	24
4 MOTORY	25
4.1 SERVO MOTORY.....	25
4.2 KROKOVÉ MOTORY.....	26

5	MOBILNÉ TECHNOLOGIE.....	28
5.1	ANDRIOD.....	28
5.1.1	Vývoj aplikácií pre Android.....	28
5.1.2	Senzory v Androidoch.....	29
5.1.3	Akcelerometer.....	29
5.1.4	Magnetometer.....	30
5.2	iOS.....	30
5.2.1	Vývoj aplikácií pre iOS.....	31
6	ANDRIOD STUDIO.....	32
6.1	JAVA.....	33
6.2	KOTLIN.....	33
7	PYTHON.....	35
7.1	GPIOZERO.....	35
II	PRAKTICKÁ ČÁST.....	37
8	MOBILNÁ APLIKÁCIA.....	38
8.1	DIZAJN.....	38
8.2	MAINACTIVITY.JAVA.....	39
9	SERVER.....	45
10	HOTOVÝ PROJEKT.....	47
10.1	KONŠTRUKCIA.....	48
10.2	FUNKČNOSŤ.....	49
11	POROVNANIE ALTERNATÍVNEJ TECHNOLOGIE.....	50
	ZÁVĚR.....	51
	SEZNAM POUŽITÉ LITERATURY.....	52
	SEZNAM OBRÁZKŮ.....	55
	SEZNAM PŘÍLOH.....	56

ÚVOD

V ére rýchlo sa rozvíjajúcich technológií sa embedded systémy stávajú neodmysliteľným nástrojom v rôznych aplikáciách automatizácie a diaľkového ovládania. Táto bakalárska práca sa zaoberá vývojom automatizovaného systému na ovládanie náklonu roviny, ktorý je integrovaný a ovládaný prostredníctvom chytrého telefónu na platforme Android, kde sa využíva senzorický systém telefónu na získavanie dát o náklone.

Teoretická časť práce obsahuje literárnu rešerš o embedded systémoch. Popisuje tiež programovanie aplikácií pre Android aj iOS. V práci bola vytvorená aplikácia pre Android, ktorá využíva senzorický systém chytrého telefónu na zber a spracovanie vstupných dát pre riadenie náklonu. V praktickej časti práce je prevedený výber a modifikácia komerčne dostupnej naklonenej roviny a je vyvinutá aplikácia, ktorá ovláda náklon na základe polohy telefónu.

Cieľom práce je nielen demonštrovať technické riešenie pre diaľkové ovládanie. Finálny prototyp ilustruje potenciál moderných technológií v transformácii ľudskej interakcie so strojovými systémami a rozširuje obzory ich praktickej aplikácie v každodennom živote.

I. TEORETICKÁ ČÁST

1 EMBEDDED SYSTÉMY

Embedded systémy sú populárne typy informačných a riadiacich systémov. Embedded systém je hardvérový počítačový systém, ktorý sa skladá z procesoru, pamäte a vstupných-výstupných zariadení s cieľom vykonávať určitú úlohu, buď samostatne alebo ako súčasť väčšieho celku. Jeho jadro tvorí integrovaný obvod, ktorý je navrhnutý na riešenie výpočtových operácií v reálnom čase.

Zložitosť embedded systému môže byť od jednoduchého mikrokontroléra až po súbor procesorov s pripojenými zariadeniami a sieťami. Existujú od najjednoduchších s absenciou užívateľského rozhrania až po systémy s komplexnými grafickými rozhraniami. Jeho komplexnosť závisí od konkrétnej úlohy, pre ktorú je určený.

Embedded systémy sa používajú v rôznych oblastiach, od spotrebiteľskej elektroniky až po priemysel a zdravotníctvo. Väčšina mikroprocesorov na svete je využitých práve v embedded systémoch.[1][2]

1.1 Ako fungujú embedded systémy

Embedded systémy fungujú na základe kombinácie hardvéru a softvéru, ktoré zahŕňajú mikroprocesory, pamäť, rozhrania pre vstupy a výstupy, a ďalšie komponenty. Avšak oproti bežným počítačom sa líšia v účele, dizajne, nákladoch a zapojení ľudského faktoru. Na rozdiel od typických pracovných staníc alebo serverov, embedded systémy využívajú programovateľné dosky s elektronickými obvodmi, ktoré riadia manipuláciu s dátami a komunikáciu s vonkajším prostredím prostredníctvom vstupných a výstupných zariadení. Výsledkom je tvorba výstupov pre výstupné zariadenia.[2]

1.2 Typy embedded systémov

Typy embedded systémov sa delia do štyroch hlavných kategórií, keď berieme do úvahy výkon a funkčné požiadavky. Patria sem: samostatné, real-time, sieťové a mobilné embedded systémy.[2]

1.2.1 Samostatné embedded systémy

Samostatné embedded systémy fungujú samostatne bez potreby podpory od druhého počítača alebo procesora. Stačí im len prijímať vstupné údaje a na základe toho a poskytujú výstup, ktorý môže byť zobrazený prostredníctvom pripojeného zariadenia. Môžu prijímať vstup v digitálnej alebo analogovej forme. Prístroje ako MP3 prehrávače, digitálne hodinky alebo kamery, sú príklady samostatných embedded systémov. Fungujú samostatne.[3]

1.2.2 Real-Time Embedded Systémy

Tento typ embedded systému potrebuje poskytovať výstupy čo najrýchlejšie v určitom časovom rámci. Preto sa často využíva v odvetviach, kde je čas dôležitý, ako sú napríklad doprava, výroba alebo zdravotníctvo. Všetky tieto odvetvia potrebujú presné a čo najrýchlejšie procesy na vykonávanie úloh.

Real-time embedded systémy sa ďalej delia na "soft" a "hard" kategórie. "Soft" kategória znamená, že systém nemusí prísne dodržiavať určený časový limit. Inak povedané, ak úloha nie je vykonaná včas, výstup sa stále považuje za prijateľný.

Medzi Soft real-time embedded systémy môžeme zaradiť nástroje na monitorovanie teploty alebo vlhkosti. V prípade oneskoreného údaju o teplote v reálnom čase získanie informácií o niečo neskoršie nebude vadit', pretože sa stále bude považovať za užitočné, pretože čas tu nie je kritickým faktorom.

Naopak, v prípade hard real-time embedded systémov sú časové obmedzenia extrémne prísne. Pri nedodržaní časového limitu môžu nastať vážne následky, ako je napríklad poškodenie zariadení, v horšom prípade zranenie ľudí alebo iné katastrofálne dôsledky.

V prípade hard real-time embedded systémov, ako napríklad riadiace stredisko na ovládanie letov, aj malé oneskorenie pri získavaní údajov môže mať katastrofálne následky, pretože na základe zastaraných informácií by sa pilot mohol zle rozhodnúť, čo by mohlo mať dopad na bezpečnosť letu, životy pasažierov a posádky.[3][4]

1.2.3 Sieťové embedded systémy

Sieťové embedded systémy používajú káblové alebo bezdrôtové siete na vytváranie výstupov. Tieto systémy často využívajú bežné procesory a obsahujú rôzne časti ako senzory, riadiace jednotky a podobne.

Príklady sieťových embedded systémov sú napríklad domáce alebo kancelárske bezpečnostné systémy, pokladňové systémy alebo bankomaty. Tieto systémy využívajú ostatné zariadenia v sieti na vykonávanie svojich funkcií. Napríklad bezpečnostný systém domu môže obsahovať senzory a kamery na monitorovanie, ktoré dokážu upozorniť majiteľa v prípade pohybu.[3]

1.2.4 Mobilné embedded systémy

Mobilné embedded systémy sú kompaktné zariadenia, ktoré sú určené na jednoduché prenášanie. Napriek tomu, že sú väčšinou kvôli tomu limitované funkciami, patria medzi populárne. Medzi mobilné embedded systémy môžeme považovať napríklad digitálne fotoaparáty alebo kamery.[3][4]

2 RASPBERRY PI

Raspberry Pi je malý a cenovo dostupný počítač, ktorý má približne rovnakú veľkosť ako kreditná karta a môže byť pripojený k hardvérovým zariadeniam, ako sú napríklad monitor, televízor, myš alebo klávesnica. Raspberry Pi sa síce nemôže rovnať dnešným výkonným stolovým počítačom ale za to, čo všetko dokáže, môžeme Raspberry v praxi považovať za plnohodnotný počítač s veľmi nízkou cenou.

Raspberry Pi sa prvý krát objavilo na trhu v roku 2012 a od tej doby vzniklo viacero verzií a variácií. Prvá generácia Raspberry Pi mala len jednoprocesorový 700 MHz CPU a len 256 MB veľkú RAM, zatiaľ čo najnovší a najlepší model má štvorprocesorový CPU s frekvenciou cez 2,4 GHz a 8 GB veľkú RAM. Cena Raspberry Pi vždy sa vždy držala pod hranicou 100 dolárov. Pi Zero, najlacnejšia verzia Raspberry Pi, je dnes dostupná za cenu do 20 eur.[6][7]

Všetky Raspberry Pi používajú operačný systém Linux. Linux je operačný systém s otvoreným zdrojovým kódom(open source), ktorý funguje ako rozhranie medzi hardvérom počítača a softvérovými aplikáciami. Python je programovací jazyk, ktorý sa často používa v Raspberry Pi. Je univerzálny a jednoduchý na pochopenie, čo ho robí ideálnym na vývoj rôznych aplikácií, ako sú grafické užívateľské rozhrania, webové stránky a webové aplikácie. Výhodou je, že aj keď človek nemá predošlé skúsenosti s Linuxom alebo s Pythonom, môže sa rýchlo naučiť, pretože učenie je jednou z hlavných charakteristík Raspberry Pi.[8]

V tomto projekte sa konkrétne pracuje s verziou *Raspberry Pi 5 4GB RAM*



Obrázok 1 Raspberry Pi 5 4GB RAM

2.1 Modely Raspberry Pi

Od počiatku bolo mnoho generácií modelov Raspberry Pi od Pi 1 až po 5, a od Pi Zero až po Pi 400. Vo väčšine generácií býval model A a model B. Model A bol dostupnejšiou a lacnejšou variantou ako model B a zvyčajne mal menej pamäte RAM a menej portov ako napríklad USB a Ethernet. Dnešný najnovší model Raspberry Pi 5 sa delí len veľkosťou pamäte RAM a to 4GB a 8GB.[7]

- Pi 1 Model B (2012)
- Pi 1 Model A (2013)
- Pi 1 Model B+ (2014)
- Pi 1 Model A+ (2014)
- Pi 2 Model B (2015)
- Pi Zero (2015)
- Pi 3 Model B (2016)
- Pi Zero W (2017)
- Pi 3 Model B+ (2018)
- Pi 3 Model A+ (2019)
- Pi 4 Model A (2019)
- Pi 4 Model B (2020)
- Pi 400 (2021)
- Pi Model 5 (2023)

2.2 Raspberry Pi 5

Raspberry Pi 5 je najnovší model Raspberry a kvôli rýchlejšiemu CPU, GPU a RAM je až 3x rýchlejší ako jeho predchodca. Okrem zvýšenia rýchlosti ponúka po prvýkrát aj RTC (real-time clock).[9]

Prehľad základných parametrov:

- **Čip (SoC):** Broadcom BCM2712
- **Processor (CPU):** ARM Cortex-A76 (ARM v8) 64-bit

- **Grafika (GPU):** VideoCore VII (800 MHz)
- **Operačná pamäť (RAM):** 4/8 GB LPDDR4X (4267 MHz)
- **WiFi:** IEEE 802.11b/g/n/ac (2.4 GHz/5 GHz)
- **Bluetooth:** Bluetooth 5.0, BLE
- **Ethernet:** Gigabit Ethernet
- **USB:** 2x USB-A 3.0 (5 Gbit/s) a 2x USB-A 2.0
- **PCI Express:** 1x PCIe 2.0
- **GPIO:** Standard 40-pin GPIO header
- **Video:** 2x micro-HDMI ports (4K60)
- **Multimedia:** H.265 (4K60 decode)
- **SD card:** microSD

2.3 Architektúra ARM

ARM je skratkou pre pokročilý počítač s redukovanou sadou inštrukcií (RISC) a predstavuje kolekciu architektúr procesorov. Jednou z hlavných výhod architektúry ARM je jej schopnosť poskytnúť vysoký výkon pri nízkej spotrebe energie, čo je ideálne pre embedded systémy, mobilné zariadenia, a IoT zariadenia. ARM procesory dokážu dlhšie fungovať na batérie a ponúkajú rýchlejší a plynulejší výkon v porovnaní s inými architektúrami. Ďalšou výhodou je, že architektúra ARM je široko akceptovaná a podporovaná priemyslom, čo znamená, že existuje veľký trh a komunita pre produkty a služby založené na ARM. Tieto procesory sa využívajú v rôznych zariadeniach vrátane smartfónov, tabletov, počítačov, serverov a ďalších. Okrem toho existuje mnoho nástrojov a platforiem podporujúcich vývoj pre ARM, ako napríklad Raspberry Pi, Arduino, Linux, Android, iOS a Windows.[10]

2.3.1 BCM2712

Broadcom BCM2712 je čip, ktorý tvorí jadro Raspberry Pi 5. Je to vylepšená verzia BCM2711, ktorý bol použitý v Raspberry Pi 4, a má mnoho spoločných vlastností s čipmi v rodine BCM27xx, ktoré boli použité v predchádzajúcich verziách Raspberry Pi.

Tento čip obsahuje štvorjadrový klastrový procesor Arm Cortex-A76, ktorý môže dosiahnuť frekvenciu až 2,4 GHz. Integrovaný grafický procesor VideoCore VII s 12 jadrami a vylepšenými funkciami pre zobrazovanie videa, HDMI kontrolér pre dva 4kp60 displeje a iné vylepšenia.

LPDDR4X zabezpečuje vysokú rýchlosť pre prenos dát, zatiaľ čo PCI Express rozhrania umožňujú pripojenie externých zariadení s vysokou prenosovou rýchlosťou.[11]

2.3.2 Konektivita čipu BCM2712

BCM2711 obsahuje nasledujúce periférie, ku ktorým môže ARM bezpečne pristupovať:[12]

- Časovače
- Interruptový kontrolér
- GPIO
- USB
- PCM / I2S
- DMA kontrolér
- I2C mastery
- SPI mastery
- PWM
- UARTs

2.4 GPIO

GPIO (General Purpose Input/Output) je najzákladnejším aspektom Raspberry Pi. GPIO piny môžu mať dva stavy, zapnuté alebo vypnuté. Môžu mať smer na prijímanie alebo odosielanie prúdu (vstup, výstup). Pomocou programovacích jazykov ako Python môžeme riadiť stav a smer GPIO pinov.

GPIO piny majú aj iné funkcie, ktoré im umožňujú komunikovať s rôznymi druhmi zariadení. Napríklad, piny GPIO3 a GPIO4 môžu slúžiť ako komunikačné piny pre I2C, SPI alebo UART, čo umožňuje pripojiť rôzne zariadenia. Ak by sme chceli využiť komunikáciu cez tieto protokoly, museli by sme ich funkcie povoliť v Raspberry Pi konfigurácií.

[13]



Obrázok 2 40-Pinový header na Raspberry Pi[13]

2.4.1 I2C

I2C je sériový protokol s nízkou rýchlosťou, ktorý umožňuje pripojiť zariadenia pomocou I2C. Zariadenia, ktoré používajú tento protokol, fungujú vo vzťahu master-slave. V I2C môže existovať viacero masterov, no každé slave zariadenie musí mať unikátnu adresu. To znamená, že na jednom I2C spojení môžeme komunikovať s viacerými zariadeniami, pretože každé z nich je jedinečné a identifikovateľné pomocou počítača a pomocou príkazov.

I2C má dva typy pripojenia: SDA a SCL. SDA posieľa a prijíma údaje, zatiaľ čo rýchlosť riadi pin SCL. I2C je rýchly a jednoduchý spôsob, ako pridať rôzne komponenty, ako sú napríklad displeje, senzory a prevodníky.

Raspberry Pi má dve I2C piny, GPIO 2 a 3 (SDA a SCL) pre I2C0 (master), Nachádzajú sa na fyzických pinoch 27 a 28. Umožňujú Raspberry Pi komunikovať s prídavnými doskami HAT (Hardware Attached on Top), ktoré sú kompatibilné.[13]

2.4.2 SPI

SPI (Serial Peripheral Interface) je ďalší protokol, ktorým sa dá pripájať kompatibilné zariadenia Raspberry Pi. Funguje podobne ako I2C, keďže medzi Raspberry Pi a pripojenými zariadeniami existuje vzťah master-slave. SPI sa bežne používa na krátke vzdialenosti na prenos dát medzi mikrokontrolérom a rôznymi komponentmi, ako sú napríklad posuvné registre, senzory a dokonca aj SD karty. Synchronizácia dát prebieha pomocou hodinového signálu (SCLK) od master zariadenia (v tomto prípade Raspberry Pi). Dáta sú následne odosielané z Raspberry na SPI komponent prostredníctvom pinu MOSI. Ak komponent potrebuje odpovedať na Raspberry Pi, posieľa dáta späť pomocou pinu MISO.[13]

2.4.3 UART

UART (Universal Asynchronous Receiver / Transmitter). Piny UART sú známe aj ako „sériové“, slúžia na komunikáciu medzi Raspberry Pi a ďalšími zariadeniami. Sú určené na odosielanie (pomocou pinu GPIO14) a prijímanie (pomocou pinu GPIO15) dát. Vďaka nim môžeme získať prístup k terminálovému prihláseniu pre bezklávesnicové nastavenie, čo znamená, že sa môžeme pripojiť k Raspberry Pi bez potreby klávesnice alebo myši.[13]

2.4.4 GND

Ground (po slovensky zem) býva označovaná ako “gnd“ alebo “-“. Ground je bod, z ktorého sa merajú všetky napätia, a taktiež uzatvára elektrický obvod. Ground je náš nulový bod, a keď pripojíme komponent, ako je LED, k zdroju napájania a ground, komponent sa stane súčasťou obvodu, cez ktorý preteká prúd a LED sa rozsvieti.

Pred pripojením akéhokoľvek komponentu je vždy rozumné začať s pripojením k GND pri zostavovaní obvodov, aby sa predišlo problémom. Okrem modelov prvej generácie A a B má Raspberry osem GND pinov. Všetky tieto piny GND sú pripojené k jednej spoločnej GROUND konekcii. Pri projekte je jedno, ktoré GND piny zvolíme.[13]

2.4.5 5V

Vďaka pinom 5V môže Raspberry Pi priamo napájať zariadenia s napätím 5V. Dokonca samotné Raspberry môže byť priamo napájané týmito pinmi. Pri práci s týmito pinmi je dôležité dodržiavať opatrnosť, pretože obchádzajú niektoré bezpečnostné prvky, ako je napríklad regulátor napätia alebo poistka.[13]

2.4.6 3V3

3V3 pin poskytuje stabilné napájanie s napätím 3,3 V, ktoré je určené na napájanie elektronických komponentov a testovanie funkčnosti LED diód. Tento pin je využívaný pri overovaní správneho pripojenia a fungovania LED diód k GPIO portom, čo umožňuje elimináciu možných hardvérových problémov v počiatočných fázach projektu. Na Raspberry Pi sa nachádzajú dva takéto piny.[13]

2.5 Wifi

Wi-Fi sieť je systém, ktorý umožňuje bezdrôtové pripojenie viacerých zariadení k internetu prostredníctvom routeru. Tento router je pripojený k internetovému modemu a slúži ako centrálny bod, z ktorého sa internetový signál vysiela na všetky zariadenia s WiFi. Toto usporiadanie poskytuje užívateľom prístup k internetu v rámci pokrytia siete, čo prináša flexibilitu a pohodlie.

Termín "Wi-Fi" bol vytvorený ako marketingový ťah s cieľom poskytnúť jednoduchý názov pre technológiu IEEE 802.11 (Institute of Electrical and Electronic Engineers), ktorá tvorí základ bezdrôtovej siete.

Wi-Fi funguje tak, že dáta sú prenášané pomocou rádiových signálov medzi routerom a zariadeniami, ako sú smartfóny, tablety alebo počítače.[14]

2.5.1 802.11b

V júli 1999 IEEE rozšíril pôvodný 802.11 štandard a vytvoril 802.11b. Ten podporuje teoretickú rýchlosť až 11 Mbps, hoci reálne dosiahnuté rýchlosti môžu byť nižšie. 802.11b využíva rovnaké rádiové frekvencie (2,4 GHz) ako pôvodný 802.11 štandard. Výrobcovia často uprednostňujú tieto frekvencie, aby znížili náklady na výrobu. Avšak zariadenia 802.11b môžu zažívať interferencie od iných zariadení, ako sú napríklad smartphony, ktoré používajú rovnaké frekvencie. Rušenie je možné minimalizovať správnym umiestnením a vzdialenosťou. Tento štandard je tiež označovaný ako Wi-Fi 1 a používa 2,4 GHz pásmo.[15]

2.5.2 802.11g

V približne rokoch 2002 a 2003 sa na trhu objavili nové zariadenia, ktoré podporovali nový štandard s označením 802.11g. Tento štandard bol vytvorený s cieľom vylepšiť výhody predchádzajúcich štandardov 802.11a a 802.11b. 802.11g umožňuje dosiahnuť teoretickej rýchlosti až do 54 Mbps a používa 2,4 GHz pásmo. Navyše je spätne kompatibilný so starším štandardom 802.11b, takže zariadenia podporujúce 802.11g budú fungovať aj s 802.11b adaptérmí a naopak. 802.11g je tiež označovaný ako Wi-Fi 3.

- **Výhody 802.11g:** Je podporovaný takmer všetkými súčasnými bezdrôtovými zariadeniami a sieťovým vybavením, čo ho robí najdostupnejšou voľbou.
- **Nevýhody 802.11g:** Ak je v sieti prítomné zariadenie 802.11b, celá sieť sa spomalí, aby sa prispôbila jeho rýchlosti. Tento štandard je navyše považovaný za najpomalší a najstarší zo všetkých stále používaných.[15]

2.5.3 802.11n

802.11n, často označovaný ako Wireless N, bol navrhnutý s cieľom zlepšiť šírku pásma oproti staršiemu štandardu 802.11g. To dosahuje pomocou viacerých bezdrôtových signálov a antén, čo sa nazýva MIMO technológia, namiesto použitia jedného signálu ako predtým. 802.11n je tiež označovaný ako Wi-Fi 4 a používa 2,4 GHz a aj 5 GHz pásma.

- **Výhody 802.11n:** Široká podpora na rôznych zariadeniach a významné zlepšenie dosahu oproti 802.11g, s teoretickou rýchlosťou až do 600 Mbps.
- **Nevýhody 802.11n:** Implementácia je nákladnejšia než pri 802.11g, použitie viacerých signálov môže interferovať s blízkymi sieťami založenými na 802.11b/g.[15]

2.5.4 802.11ac

802.11ac je štandard ktorý ako prvý začal byť populárny pre bežné používanie. Využíva technológiu duálneho pásma, čo umožňuje pripojenie zariadení na obe Wi-Fi frekvencie 2,4 GHz a 5 GHz naraz. Štandard 802.11ac je spätne kompatibilný s predchádzajúcimi štandardmi 802.11a/b/g/n a ponúka teoretickú rýchlosť až do 1300 Mbps na 5 GHz pásme a až do 450 Mbps na 2,4 GHz pásme. Väčšina bežných domácich routerov podporuje tento štandard.[15]

2.5.5 2,4 GHz

Termín 2,4 GHz označuje frekvenčné pásmo, ktoré používajú bezdrôtové routery na prenos Wi-Fi signálu. Dnes väčšina routerov pracuje v tzv. dual-band móde, čo znamená, že sú schopné vysielat'

na oboch frekvenčných pásmach naraz. Výhodou 2,4 GHz je, že má väčší dosah a omnoho lepšiu prenikavosť cez objekty a steny ako 5 GHz. Väčšinou dosahuje rýchlosti okolo 150Mbps s teoretickou maximálnou rýchlosťou za ideálnych podmienok až do 600 Mbps. Nevýhodou je, že mnoho domácich zariadení, ako sú napríklad mikrovlnné rúry alebo otvárate garážových dverí, ktoré využívajú rovnaké frekvenčné pásmo, spôsobujú rušenie a zníženie rýchlosti Wi-Fi pripojenia.[16]

2.5.6 5 GHz

Wi-Fi signál na frekvencii 5 GHz používa vyššie frekvencie ako Wi-Fi signál na 2,4 GHz. Tieto vyššie frekvencie poskytujú rýchlejšie prenosové rýchlosti a menej rušenia v porovnaní s nižšími frekvenciami na pásme 2,4 GHz. Vďaka vyššej frekvencii sú aj vyššie prenosové rýchlosti a to až do 1,300 Mbps. V 5 GHz pásme je k dispozícii až 42 kanálov, zatiaľ čo pásmo 2,4 GHz ponúka len 14 kanálov. Väčší počet kanálov umožňuje viacej rovnomerné rozloženie Wi-Fi signálov, čo znižuje riziko preplnenia siete. Wi-Fi využíva tieto kanály na vysielanie internetového signálu.[16]

2.6 Bluetooth

Bluetooth je bezdrôtová technológia, ktorá umožňuje prenos dát medzi zariadeniami na krátke vzdialenosti pomocou rádiových vln s frekvenciou 2,4 GHz. Táto technológia sa používa na pripojenie rôznych zariadení ako sú slúchadlá, smartfóny, počítače a mnoho ďalších. Bluetooth zariadenia komunikujú v rámci tzv. piconetov alebo osobných sieťových oblastí, kde viacero zariadení môže byť spárovaných a vzájomne si vymieňať informácie.

Bluetooth funguje na princípe párovania, pri ktorom sa zariadenia najprv identifikujú a navzájom si dôverujú. Na uľahčenie komunikácie bez zásahu užívateľa Bluetooth používa metódu adaptívneho skákania frekvencií, ktorá minimalizuje rušenie a zvyšuje efektivitu prenosu. Okrem toho moderné Bluetooth technológie, ako je Bluetooth Low Energy (LE), ponúkajú energeticky úsporné riešenia pre zariadenia vyžadujúce menej energie pri zachovaní komunikačného dosahu.[17]

2.6.1 BLE

Bluetooth Low Energy (BLE) je špeciálna verzia technológie Bluetooth navrhnutá pre situácie, v ktorých je dôležitá nízka spotreba energie. Táto technológia umožňuje komunikáciu medzi zariadeniami s podstatne nižšou spotrebou energie v porovnaní s klasickým Bluetooth (Bluetooth Classic). Bluetooth LE je ideálne pre zariadenia ako sú fitness náramky, smart hodinky a iné

zariadenia nositeľnej elektroniky, ktoré vyžadujú dlhodobú výdrž batérie pri sporadickom prenose malého množstva dát.

Rozdiel medzi Bluetooth LE a Bluetooth Classic spočíva hlavne v energetických požiadavkách a spôsoboch použitia. Kým Bluetooth Classic je vhodný pre zariadenia vyžadujúce nepretržitý prenos veľkých objemov dát, ako sú audio streamy (napr. pre bezdrôtové slúchadlá), Bluetooth LE je optimalizované pre príležitostný prenos malých dátových balíčkov a má oveľa menšiu spotrebu energie.[17]

3 LINUX

Linux je populárny a široko používaný operačný systém s otvoreným zdrojovým kódom (open source). Ako operačný systém, Linux funguje ako základ pre všetok softvér na počítači, spracováva žiadosti od programov a sprostredkováva ich komunikáciu s hardvérom počítača. Hlavnou výhodou Linuxu oproti iným operačným systémom, ako sú Windows alebo macOS, je práve to, že je open source. Zdrojový kód Linuxu je verejne dostupný na prezretie, úpravy, čo umožňuje užívateľom aktívne sa zapájať do rozvoja a zdokonaľovania Linuxu.

Linux je tiež mimoriadne prispôsobiteľný vďaka mnohým distribúciám, ktoré obsahujú rôzne softvérové možnosti a komponenty systému. Linux sa používa na webových serveroch, kde generuje veľkú časť internetových stránok, ale nájdeme ho aj v mnohých zariadeniach ako sú Android telefóny, tablety, digitálne úložiská a dokonca aj v automobiloch. Systém je udržiavaný a neustále vylepšovaný komunitou vývojárov a podporovaný firmami poskytujúcimi komerčnú podporu.[18]

3.1 Raspbian

Raspbian je operačný systém založený na Debiane, ktorý bol špeciálne navrhnutý pre hardware Raspberry Pi. Tento systém bol pôvodne vytvorený ako nezávislý projekt programátormi Peterom Greenom a Mikeom Thompsonom a je známy pre svoju optimalizáciu pre ARM procesory, ktoré sú srdcom zariadení Raspberry Pi. Prvýkrát bol predstavený v roku 2013 a odvtedy sa stal jedným z najpopulárnejších distribúcií Linuxu pre Raspberry Pi, ktorý je obzvlášť cenený pre jeho užívateľsky prívetivé prostredie a rozsiahlu podporu softvéru. V priebehu rokov sa Raspbian vyvinul a rozširoval, čo prinieslo množstvo aktualizácií a vylepšení funkcionality. Systém je známy svojím grafickým používateľským rozhraním a predinštalovanými aplikáciami, ktoré zahŕňajú internetové prehliadače, kancelárske aplikácie a nástroje na programovanie.

V roku 2020 prešiel Raspbian významnou zmenou, keď bol premenovaný na Raspberry Pi OS. Tento krok bol uskutočnený s cieľom zjednotiť značku a lepšie odrážať súčasné možnosti a rozsah systému, ktorý teraz podporuje aj 64-bitovú architektúru. Táto aktualizácia umožnila efektívnejšie využívanie výkonnejších modelov Raspberry Pi s väčšou pamäťou RAM.[19]

3.1.1 Debian 12

Debian 12, známý ako "Bookworm", je najnovšia verzia Debian GNU/Linux, uvedená na trh v júni 2023. Toto vydanie nahradilo predchádzajúcu verziu Debian 11 "Bullseye" a je zamerané na dlhodobú podporu (LTS), ktorá je zabezpečená na nasledujúcich päť rokov. Bookworm priniesol významné aktualizácie softvéru a systému vrátane nového Linux jadra 6.1, ktoré zlepšuje výkon a podporuje moderné hardvérové platformy. Toto vydanie Debianu rozširuje svoju knižnicu o viac ako 11,000 nových balíčkov. Významné zmeny zahŕňajú začlenenie proprietárneho softvéru priamo do inštalačných ISO súborov a predvolené použitie zvukového servera Pipewire v prostredí GNOME. Debian 12 "Bookworm" poskytuje robustné a flexibilné riešenie pre rôzne typy používateľov, od začiatočníkov až po skúsených profesionálov, vďaka svojej podpore pre množstvo hardvérových architektúr a novým verziám desktopových prostredí a aplikácií. V projekte tejto práce sa pracuje s Raspberry Pi, ktoré beží na Debiane 12 Bookworm. Príkazom "cat /etc/os-release" sa dá zistiť aktuálna verzia os.[20]

```
stanko@raspberrypi:~ $ cat /etc/os-release
PRETTY_NAME="Raspbian GNU/Linux 12 (bookworm)"
NAME="Raspbian GNU/Linux"
VERSION_ID="12"
VERSION="12 (bookworm)"
VERSION_CODENAME=bookworm
ID=raspbian
ID_LIKE=debian
HOME_URL="http://www.raspbian.org/"
SUPPORT_URL="http://www.raspbian.org/RaspbianForums"
BUG_REPORT_URL="http://www.raspbian.org/RaspbianBugs"
```

Obrázok 3 Príkaz na zistenie verzie OS

4 MOTORY

Elektrické motory, ako sú krokové a servomotory, premieňajú elektrickú energiu na mechanickú energiu a umožňujú vysokú presnosť ovládania polohy, rýchlosti a zrýchlenia. Krokové motory rozdeľujú celú rotáciu na rovnaké kroky a dokážu presne udržiavať požadovanú polohu bez použitia senzorov na spätnú väzbu, čo je ideálne pre aplikácie, kde je požadované detailné polohovanie, ako sú 3D tlačiarne alebo malé CNC stroje. Naproti tomu servomotory, sú vybavené senzormi, ktoré poskytujú neustálu spätnú väzbu o polohe, čím zaisťujú presnosť. Tieto motory sa často používajú v robotike a automatizovaných výrobných systémoch, kde je potrebná presná kontrola pohybu.

Oba typy motorov môžu byť napájané jednosmerným alebo striedavým prúdom a môžu obsahovať rôzne technológie riadenia pre optimalizáciu ich výkonu a efektivity. V dôsledku ich schopnosti presného riadenia a adaptability sa krokové a servomotory stali základnými komponentmi v moderných priemyselných aplikáciách, od montážnych línii až po pokročilé robotické systémy.[21]

4.1 Servo motory

Servo motory sú typom uzavretého regulačného systému, ktorý kombinuje motor, riadiacu jednotku a spätnoväzbový mechanizmus, ako je enkodér alebo resolver. Tieto motory sú navrhnuté pre presné riadenie polohy, rýchlosti a zrýchlenia mechanickej sústavy a sú vhodné pre široké spektrum aplikácií vrátane robotiky, CNC strojov a automatizovaných výrobných systémov, kde je dôležitá presnosť, odozva a efektívnosť.

Existujú dva hlavné typy servo motorov: striedavé servo motory (AC) a jednosmerné servo motory (DC). AC servo motory sa často používajú v priemyselných aplikáciách vďaka ich vysokej výkonnosti, rýchlosti a účinnosti. Na druhej strane, DC servo motory sa častejšie vyskytujú v menších, batériou poháňaných zariadeniach, vďaka nižším požiadavkám na výkon a jednoduchosti ovládania.

AC servo motory môžu byť synchronné alebo asynchronné. Synchronné AC servo motory udržiavajú konštantnú rýchlosť za rôznych záťaží a majú rotor, ktorý sa otáča synchronne so statorovým magnetickým poľom. Asynchronné AC servo motory, majú rotor, ktorý sa otáča mierne pomalšie než magnetické pole statora.

DC servo motory sa rozdeľujú na kefkové a bezkefkové, čo závisí od použitej metódy komutácie na riadenie prúdu v cievkach motora. Kefkové DC servo motory využívajú mechanické kefkky na prepnutie prúdu v cievkach. Naopak, bezkefkové DC servo motory používajú elektronickú komutáciu, ktorá zabezpečuje vyššiu efektívnosť, spoľahlivosť a dlhšiu životnosť oproti kefkovým modelom.

Servo motor sa skladá z niekoľkých kľúčových súčastí, ktoré spoločne pracujú na dosiahnutí presného riadenia pohybu. Tieto súčasti zahŕňajú samotný motor, ovládaciu jednotku a mechanizmus na spätnú väzbu. Motor môže byť typu AC alebo DC a je zodpovedný za konverziu elektrickej energie na mechanický pohyb. Ovládacia jednotka, známa tiež ako servo riadič alebo servo zosilňovač, interpretuje vstupné signály z ovládacieho systému a vytvára príslušné výstupné signály pre riadenie motora.

Mechanizmus spätnej väzby predstavuje nevyhnutnú časť systému každého servo motora, pretože poskytuje neustále aktualizované informácie o aktuálnej polohe, rýchlosti a krútiacom momente motora. Tieto údaje sú prenášané do ovládacej jednotky, ktorá ich neustále porovnáva s požadovanými hodnotami výkonu. Na základe tejto analýzy ovládacia jednotka prispôbuje výkon motora tak, aby eliminovala všetky prípadné nepresnosti v jeho chode. Medzi zariadenia používané na spätnú väzbu v servo motoroch patria predovšetkým rôzne typy enkodérov.[22]

4.2 Krokové motory

Krokové motory sú druhom bezkefkových elektromotorov pracujúcich na jednosmerný prúd a sú konštruované tak, aby rozdelili jednu úplnú otáčku na rovnaké časti (kroky). Tento motor umožňuje veľmi presné riadenie polohy bez potreby spätnej väzby, pretože každý pohyb motora je presne definovaný počtom impulzov dodaných ovládacím jednotkám, čo umožňuje motoru pohybovať sa v pevných krokoch.

Krokové motory sú ideálne pre aplikácie, kde je požadovaná vysoká presnosť polohovania, ako sú CNC stroje, 3D tlačiarne a robotické zariadenia. Tieto motory sú obľúbené vďaka svojej schopnosti presného riadenia a jednoduchosti použitia.

Existujú tri hlavné typy krokových motorov: motor s trvalými magnetmi (PM), motor s variabilnou reluktanciou (VR) a hybridný motor. Motor s trvalými magnetmi obsahuje rotor s trvalými magnetmi a stator s elektromagnetickými cievkami. Aktiváciou cievok v špecifickom poradí sa vytvorí magnetické pole, ktoré interaguje s polom rotora a spôsobuje jeho rotáciu v krokoch. Motor s variabilnou reluktanciou má rotor vyrobený z mäkkého magnetického materiálu a neobsahuje trvalé magnety. Statorové cievky vytvárajú magnetické pole, ktoré priťahuje zuby rotora a tým spôsobuje jeho krokový pohyb. Hybridný motor kombinuje vlastnosti oboch predchádzajúcich typov a poskytuje vyšší krútiaci moment a lepšie výkonnostné charakteristiky.

Základné komponenty krokového motora zahŕňajú rotor, stator a vinutia motora. Rotor, ktorý je rotujúcou časťou motora, môže obsahovať trvalé magnety alebo mäkký magnetický materiál v

závislosti od typu motora. Stator, ktorý je statickou časťou motora, pozostáva zo série elektromagnetických cievok usporiadaných okolo obvodu motora.

Krokové motory sa rozlišujú na bipolárne a unipolárne podľa usporiadania ich vinutí. Bipolárne motory sú vybavené dvoma vinutiami pre každú fázu, pričom každé vinutie je napojené na vlastný zdroj energie. Toto usporiadanie poskytuje vyšší krútiaci moment, avšak vyžaduje zložitejšiu riadiacu elektroniku. Na druhej strane, unipolárne motory obsahujú jedno vinutie s centrálnym odbočením na fázu, čo značne zjednodušuje riadiacu elektroniku, ale zároveň môže mať za následok nižší krútiaci moment v porovnaní s bipolárnymi motormi.[22]

5 MOBILNÉ TECHNOLOGIE

5.1 Andrioid

Android, operačný systém s otvoreným zdrojovým kódom (open source) vyvíjaný spoločnosťou Google, bol prvýkrát uvedený na trh v roku 2008. Je založený na jadre Linuxu a je dnes najpoužívanejším mobilným operačným systémom na svete s viac ako 3 miliardami aktívnych zariadení. Android je známy svojou vysokou úrovňou prispôsobivosti, ktorá užívateľom umožňuje zmeniť vzhľad a funkcie ich zariadení pomocou rôznych spúšťačov, widgetov, tém a aplikácií. Tento systém podporuje širokú škálu hardvéru od rôznych výrobcov, ako sú Samsung, Huawei, Xiaomi, OnePlus a Motorola, čo ponúka užívateľom rozmanité možnosti výberu zariadenia, ktoré vyhovuje ich potrebám a rozpočtu.

Android je tiež integrovaný s množstvom služieb a produktov spoločnosti Google, vrátane Gmailu, Google Fotiek, Google Map, Google Assistant, Google Play Store a ďalších. Vďaka svojej otvorenosti a dostupnosti na rôznych zariadeniach, Android pokračuje v dominancii na trhu mobilných operačných systémov, čo odráža jeho schopnosť prispôbiť sa rôznym požiadavkám a preferenciám užívateľov.[23]

5.1.1 Vývoj aplikácií pre Android

Android poskytuje vývojárom široké možnosti. Vďaka tomu môžu vývojári pristupovať k zdrojovému kódu, upravovať ho a prispôbovať hardvéru konkrétnych zariadení. Táto flexibilita umožňuje vývojárom navrhovať aplikácie, ktoré sú plne optimalizované pre rôzne typy zariadení od rôznych výrobcov.

Sada nástrojov pre softvérový vývoj, známa ako Android SDK, je neoceniteľnou súčasťou Android Studio, ale je dostupná aj samostatne pre použitie v iných integrovaných vývojových prostrediach (IDE). Tieto nástroje znižujú redundanciu kódu a zabezpečujú, že aplikácia zaberie menej miesta a rýchlejší chod.

Android sa neustále vyvíja vďaka rozsiahlej komunite vývojárov, ktorí pravidelne poskytujú spätnú väzbu na zlepšenie operačného systému. Toto vedie k pridávaniu nových funkcií a odstraňovaniu chýb. Rýchla distribúcia aplikácií v obchode Google Play je ďalšou výhodou.

Medzi hlavné nevýhody vývoja pre Android patrí dlhší čas vývoja aplikácií. Vývojári väčšinou používajú Javu alebo Kotlin, ktoré vyžadujú písanie dlhších kódov oproti Swiftu, používanému pri vývoji pre iOS. Okrem toho musia vývojári testovať aplikácie na rôznych zariadeniach a verziách operačného systému, čo predlžuje celý proces vývoja.

Bezpečnosť je ďalšou značnou výzvou. Android zariadenia sú náchylné na bezpečnosť. Zabezpečenie dostatočnej ochrany pred hrozbami je pre vývojárov Android aplikácií kľúčovou úlohou.[24]

5.1.2 Senzory v Androidoch

Senzory sú integrované súčasťou väčšiny zariadení s Androidom a umožňujú merať rôzne pohybové, orientačné a environmentálne parametre. Tieto senzory poskytujú presné surové dáta, ktoré možno využiť na monitorovanie trojrozmerného pohybu zariadenia alebo zmien v okolitom prostredí.

Android podporuje tri hlavné kategórie senzorov:

- **Pohybové senzory:** Tieto senzory zaznamenávajú silu pohybu a rotácie na troch osiach. Zahŕňajú akcelerometre, gravitačné senzory, gyroskopy a senzory rotačného vektora.
- **Environmentálne senzory:** Merajú rôzne environmentálne faktory ako teplotu vzduchu, tlak, osvetlenie a vlhkosť. Patria sem barometre, fotometre a teplomery.
- **Pozičné senzory:** Tieto senzory zisťujú fyzickú polohu zariadenia. V tejto kategórii nájdeme orientačné senzory a magnetometre.

Senzory v Android SDK poskytujú triedy a rozhrania, ktoré umožňujú prístup k týmto senzorom a spracovanie ich dát. S ich pomocou sa dá zistiť, ktoré senzory sú na zariadení dostupné, zistiť ich špecifikácie a implementovať detektory na zmeny stavu, ktoré budú monitorovať zmeny a získavať orientačné údaje.

Dostupnosť senzorov sa môže líšiť od zariadenia k zariadeniu a taktiež sa môže meniť v rôznych verziách Androidu.[25]

5.1.3 Akcelerometer

V mnohých zariadeniach so systémom Android sa nachádza senzor nazývaný akcelerometer. Tento senzor dokáže zaznamenávať silu pohybu zariadenia pozdĺž troch osí: x, y a z. Tieto merania zahŕňajú aj vplyv zemskej gravitácie. Vďaka akcelerometru je možné detekovať a interpretovať rôzne typy pohybov zariadenia, ako sú otáčanie, potras alebo naklonenie.

Využitie akcelerometra je široké, od fitness aplikácií, ktoré sledujú pohybovú aktivitu, až po hry, kde sa pomocou nakláňania ovládajú herné prvky. Akcelerometre sú tiež nevyhnutné v aplikáciách pre navigáciu alebo bezpečnostné systémy, kde je dôležité zaznamenávať a reagovať na rýchle zmeny v pohybe zariadenia.

S rozhraním senzorov na platforme Android môžu vývojári ľahko pristupovať k funkciám akcelerometra. Je možné overiť, aké senzory sú na zariadení dostupné, získať detailné informácie o každom senzore, vrátane jeho maximálnej možnej meranej hodnoty, rozlíšenia a spotreby energie. [25]

5.1.4 Magnetometer

Magnetometer je ďalší užitočný senzor, ktorý sa nachádza v niektorých Android zariadeniach. Tento hardvérový senzor je schopný merať silu magnetického poľa okolo zariadenia v troch fyzických osiach: x, y a z. Údaje sú vyjadrené v mikroteslách (μT) a umožňujú zariadeniu zistiť jeho orientáciu voči Zemi.

Hlavné využitie magnetometra je v navigačných aplikáciách, kde poskytuje kompasové údaje, čo je kľúčové pre určovanie smeru. Tento senzor je tiež využívaný v aplikáciách, ktoré potrebujú rozpoznávať a reagovať na zmeny v magnetickom poli zariadenia.

Android poskytuje rozhranie na prístup k funkcionalite magnetometra, čo umožňuje vývojárom zisťovať dostupnosť tohto senzora na konkrétnom zariadení, ako aj získať podrobné informácie o jeho vlastnostiach. To zahŕňa maximálny rozsah merania, rozlíšenie a energetickú náročnosť senzora. Vďaka týmto údajom môžu vývojári optimalizovať svoje aplikácie pre rôzne zariadenia, čím zabezpečia ich správnu funkčnosť a efektívnosť. [25]

5.2 iOS

iOS je exkluzívny operačný systém vyvíjaný spoločnosťou Apple, založený na jadre Darwin, ktoré má unixový základ. Uvedený na trh v roku 2007, systém iOS sa rýchlo stal druhým najviac rozšíreným mobilným operačným systémom na svete, s používateľskou základňou presahujúcou miliardu aktívnych zariadení. Charakterizuje ho čisté a intuitívne používateľské rozhranie, konzistentný výkon a hlboká integrácia s produktmi a službami firmy Apple.

Začlenenie služieb ako iCloud, iMessage a FaceTime, spolu s aplikáciami ako Apple Music a Apple Pay, umožňuje používateľom iOS zariadení vychutnať si bezproblémovú synchronizáciu a funkčnosť naprieč celým ekosystémom zariadení Apple.

Naviac, App Store poskytuje bohatú zbierku aplikácií, ktoré sú dôkladne preskúmané, aby zabezpečili ich bezpečnosť a súlad s vysokými štandardmi firmy Apple.

iOS sa vyznačuje prísnyimi bezpečnostnými protokolmi vrátane šifrovania a biometrickej autentifikácie, čo posilňuje jeho povest' ako jednej z najbezpečnejších mobilných platforiem dostupných dnes. iOS je optimalizovaný výlučne pre produkty. Táto exkluzivita zaručuje

konzistentnosť a pravidelnosť softvérových aktualizácií naprieč všetkými zariadeniami. Tento cieľový prístup zabezpečuje, že každé zariadenie s iOS poskytuje stabilný a plynulý užívateľský zážitok.[23]

5.2.1 Vývoj aplikácií pre iOS

Vývoj aplikácií pre iOS predstavuje atraktívnu príležitosť vzhľadom na rozšírenosť a popularitu zariadení Apple. iOS ponúka vývojárom významné výhody, vďaka ktorým môžu vytvárať inovatívne a hodnotné aplikácie pre milióny používateľov po celom svete. Hlavné výhody vývoja pre iOS zahŕňajú rýchlejší proces, ktorý je umožnený vďaka použitiu jazyka Swift. Tento programovací jazyk je menej náročný na písanie kódu v porovnaní s Javou alebo Kotlinom, ktoré sú bežné pre Android. Swift zjednodušuje proces vývoja. Ďalšou výhodou je kratší testovací proces. Keďže iOS používajú len zariadenia Apple, vývojári potrebujú testovať aplikácie na menej zariadeniach a väčšina Apple užívateľov preferuje najnovšiu verziu operačného systému a to zjednodušuje proces testovania.

Medzi hlavné nevýhody patrí časovo náročný proces schvaľovania aplikácií v App Store kvôli prísnyim pravidlám. Apple kladie vysoké nároky na dodržiavanie špecifických usmernení a pri ich nedodržaní môže zamietnuť aplikácie, čo obmedzuje možnosti prispôsobenia a experimentovania s novými návrhmi.[24]

6 ANDRROID STUDIO

Android Studio je oficiálne integrované vývojové prostredie (IDE) navrhnuté pre platformu Android, ktoré poskytuje komplexné nástroje pre celý vývojový cyklus aplikácií. Vďaka svojej širokej funkcionalite a podpore mnohých nástrojov a knižníc sa stalo neoddeliteľnou súčasťou procesu vývoja aplikácií pre Android, čo ho robí ideálnym riešením pre vývojárov rôznych úrovní skúseností. Android Studio bolo navrhnuté spoločnosťou Google a poskytuje integrované prostredie, ktoré umožňuje efektívny a systematický vývoj aplikácií. Vďaka vizuálnym nástrojom a emulátoru pomáha zjednodušovať a zefektívňovať procesy ako sú návrh, implementácia, testovanie a nasadzovanie aplikácií.

Základné prvky Android Studia:

- **Android SDK a AVD Manager:** Tieto nástroje sú kľúčové pre vývoj, umožňujú vývojárom prístup k najnovším API a pomáhajú pri tvorbe a správe virtuálnych Android zariadení pre testovanie aplikácií.

- **Integrované nástroje na ladenie a profilovanie:**

Android Studio je vybavené výkonnými nástrojmi na ladenie a profilovanie, ktoré vývojárom umožňujú identifikovať a riešiť problémy s výkonom a stabilitou aplikácií.

- **Podpora pre viaceré programovacie jazyky:**

Hlavné jazyky podporované v Android Studiu sú Java a Kotlin, ktoré sú široko používané pri vývoji Android aplikácií. Okrem toho prostredie podporuje aj použitie C++ a NDK (Native Development Kit) pre komponenty vyžadujúce vysoký výkon.

Významné súčasti Android Studia:

- **Súbor AndroidManifest.xml:** Tento manifestový súbor je nevyhnutný pre každú Android aplikáciu, obsahuje dôležité informácie o aplikácii, ako sú povolenia, komponenty (aktivity, služby, prijímače) a verziu aplikácie.
- **Konfiguračný súbor Gradle (build.gradle):** Tento súbor umožňuje definovať nastavenia kompilácie projektu, vrátane závislostí, verzii SDK a okrem toho tento súbor obsahuje aj nastavenia pre digitálne podpisovanie aplikácie, čo je kritický krok pred jej distribúciou na oficiálne distribučné platformy, ako je Google Play.
- **Version control systém Git:** Git je integrovaný priamo v Android Studiu, čo umožňuje vývojárom efektívne spravovať zmeny v kóde.

- **Nástroje pre ladenie aplikácií:** Android Studio obsahuje rozsiahle nástroje pre ladenie, ktoré vývojárom umožňujú monitorovať a analyzovať správanie aplikácie v reálnom čase a identifikovať problémy s výkonom alebo chybami v kóde.

Android Studio je neoceniteľným nástrojom pre vývojárov Android aplikácií, ponúka bohaté prostredie pre kódovanie, testovanie a nasadzovanie aplikácií. S jeho pomocou môžu vývojári efektívne vytvárať aplikácie, ktoré sú optimalizované pre rozličné Android zariadenia a verzie operačného systému, čím zabezpečujú širšiu dostupnosť a lepšiu užívateľskú skúsenosť.[26]

6.1 Java

Java je viacplatformový a objektovo orientovaný programovací jazyk, ktorý beží na miliardách zariadení po celom svete, je stále najpopulárnejším jazykom pre vývojárov aplikácií, aj keď bol vytvorený už pred viac ako 20 rokmi. Java je programovací jazyk, ktorý sa využíva vo veľkom rozsahu aplikácií od webových stránok až po mobilné aplikácie a cloudové služby. Vývojári obľubujú Javu pre jej univerzálnosť a schopnosť bežať na rôznych platformách bez potreby zmeny kódu, čo sa v odbornej terminológii označuje ako WORA(Write Once Run Anywhere). Táto schopnosť je obzvlášť cenná pri vytváraní aplikácií pre cloud, kde môžu aplikácie bežať na rôznych serveroch bez ohľadu na ich operačný systém.

Java sa tiež vyznačuje rozsiahlym využitím v oblastiach ako sú veľké dáta a umelá inteligencia, kde je veľmi efektívna na spracovanie veľkých objemov dát. Vývojári využívajú bohaté knižnice a API a vďaka nim je Java obľúbený výber pre nových aj skúsených programátorov.

Java Virtual Machine (JVM) je dostupná pre väčšinu hlavných operačných systémov vrátane Windowsu, macOS a Linuxu a je kľúčovým prvkom, ktorý umožňuje Java aplikáciám fungovať na rôznych platformách bez nutnosti prepisu kódu. JVM funguje ako interpretér medzi bajtkódom, do ktorého sú Java programy kompilované a hardvérom počítača, na ktorom program beží. Tento proces, známy ako (WORA), zabezpečuje, že Java kód napísaný na jednom systéme môže byť spustený na ľubovoľnom inom systéme, ktorý má nainštalovaný JVM.[27][28]

6.2 Kotlin

Kotlin je open-source programovací jazyk, navrhnutý tak, aby plne spolupracoval s Androidom a JVM(Java Virtual Machine). V porovnaní s Javou ponúka Kotlin objektovo orientované a funkcionálne programovacie prvky so stručnejšou syntaxou a vlastnosťami null safety. Táto stručnosť kódu umožňuje vývojárom dosahovať rovnaké výsledky s menším počtom riadkov, čo zlepšuje jeho čitateľnosť.

Podobne ako Java, Kotlin je staticky typovaný jazyk, ktorý vykonáva kontroly typov premenných počas kompilačnej fázy, čo zvyšuje ochranu pred chybami a zlepšuje výkon. Kotlin bol pôvodne navrhnutý pre JVM, ale umožňuje aj kompiláciu do JavaScriptu alebo natívnych binárnych súborov, ktoré môžu bežať priamo na rôznych operačných systémoch bez JVM. Toto rozšírenie umožňuje vývojárom používať Kotlin pre širokú škálu aplikácií vrátane webových a desktopových aplikácií.

Napriek mnohým výhodám má Kotlin aj nevýhody, napríklad v prípade clean buildov je kompilácia pomalšia v porovnaní s Javou. Avšak pri inkrementálnych buildov je Kotlin výrazne rýchlejší vďaka princípom, ako sú: vyhýbanie sa kompilácii a inkrementálna kompilácia, ktoré znižujú potrebu častého prekompilovania kódu.

Kotlin, ako výtvor Dmitrija Jemerova, bol inšpirovaný funkčnými vlastnosťami jazykov Scala a Groovy, ale s cieľom dosiahnuť rýchlejšiu kompiláciu podobnú Jave. Tento jazyk tak stavia most medzi modernými programovacími prístupmi a efektívnou realizáciou, čo ho robí ideálnou voľbou pre súčasných i budúcich vývojárov softvéru.[29]

7 PYTHON

Python je vysoko flexibilný a všeobecne používaný programovací jazyk, ktorý nachádza uplatnenie v mnohých oblastiach od vývoja webových aplikácií až po machine learning. Vďaka svojej jednoduchosti a čitateľnosti kódu je ideálny pre začiatočníkov aj pokročilých vývojárov. Tento jazyk používa rozsiahlu knižnicu modulov a balíčkov, ktoré umožňujú rýchle a efektívne riešenie rozličných programátorských úloh.

Vývoj webových aplikácií je ďalšou dominantnou oblasťou, kde Python zohráva kľúčovú rolu, najmä pri vývoji serverových častí aplikácií pomocou frameworkov ako Django alebo Flask. Tieto nástroje umožňujú efektívnu prácu s databázami, spracovanie dát a zabezpečenie komunikácie medzi serverom a klientom.

Automatizácia opakujúcich sa úloh je tiež oblasť, kde Python exceluje. Umožňuje užívateľom efektívne spracovávať súbory, automatizovať administratívne úlohy a integrovať rôzne systémové procesy, čím šetrí čas a znižuje možnosť chýb.

Napriek svojej jednoduchosti, Python poskytuje výkonné nástroje pre softvérové testovanie a prototypovanie, čo umožňuje vývojárom rýchlo testovať a iterovať na svojich projektoch. Vďaka komunitě a obrovskej základni modulov a knižníc môžu programátori rýchlo nájsť riešenia a nové techniky pre svoje projekty. Python pomenovaný po Monty Pythonovi, je dnes vďaka svojej flexibilitě a prístupnosti jedným z najpoužívanejších jazykov vo svete programovania.[30]

7.1 Gpiozero

Modul GPIO Zero je navrhnutý pre zjednodušenie interakcie s hardvérom prostredníctvom Raspberry Pi, poskytuje vysokú úroveň abstrakcie, čím umožňuje užívateľom ľahko ovládať rôzne komponenty ako LED, tlačidlá, motory a ďalšie zariadenia pomocou jednoduchých príkazov Pythonu. Jeho použitie je intuitívne aj pre začiatočníkov, vďaka čomu sa stáva populárnym nástrojom v projektoch.

Napríklad, pre ovládanie LED diódy sa používa rozhranie LED, kde sa môže jednoducho nastaviť zapnutie alebo vypnutie diódy alebo jej blikanie. Rovnako ľahko sa ovláda aj tlačidlo, s možnosťou priradiť rôzne funkcie jeho stlačeniu alebo uvoľneniu. GPIO Zero taktiež podporuje zložitejšie zariadenia ako sú kamery, senzory pohybu alebo teploty, ktoré umožňujú realizáciu sofistikovanejších projektov bez potreby hlbokých znalostí o elektronike.

Jeho výhodou je tiež podpora pre viacero spôsobov číslovania pinov, čo zjednodušuje prechod od iných knižníc alebo platform. Modul poskytuje bohatú dokumentáciu, čo ďalej podporuje jeho využívanie v praxi.

Vývojári môžu ďalej rozširovať možnosti GPIO Zero prispôbovaním existujúcich tried alebo pridávaním nových, čo umožňuje jeho použitie vo veľmi širokom spektre projektov. Celkovo GPIO Zero značne znižuje bariéru pre vstup do sveta hardvérového programovania, robí experimenty s Raspberry Pi dostupnejšími.[31]

II. PRAKTICKÁ ČÁST

8 MOBILNÁ APLIKÁCIA

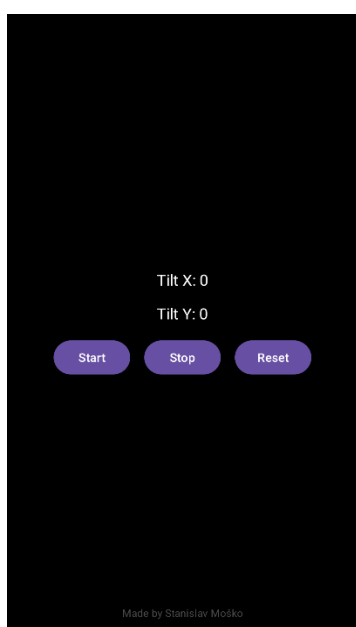
8.1 Dizajn

V rámci dizajnu mojej Android aplikácie bol kladený dôraz na vytvorenie jednoduchého a intuitívneho užívateľského rozhrania, aby bolo používateľom čo najľahšie orientovať sa a efektívne využívať funkcionality aplikácie. Cieľom bolo minimalizovať zložitosť interakcií a zabezpečiť, aby rozhranie bolo čo najjednoduchšie pre všetkých užívateľov.

Rozhranie aplikácie je založené na kombinácii RelativeLayout a LinearLayout, čo umožňuje flexibilné a efektívne rozloženie prvkov. UI komponenty, ako sú tlačidlá a textové polia, sú organizované s dôrazom na ergonómiu a logickú štruktúru. Užívateľské rozhranie neobsahuje zbytočné elementy alebo tlačidlá, ktoré by mohli užívateľa zmiatať alebo zbytočne komplikovať jeho interakcie s aplikáciou.

Pre dizajn som použil štandardizované ikony a farby z Android material design knižnice, zabezpečujúce vizuálnu príťažlivosť. Tieto designové prvky nielenže prispievajú k estetickému príjemnosti aplikácie, ale tiež podporujú užívateľsku orientáciu pomocou zavedených vizuálnych štandardov, ktoré sú bežné a dobre známe medzi užívateľmi.

V súbore activity_main.xml boli definované jednotlivé UI komponenty a ich rozloženie, zabezpečujúc, že rozhranie bude reagovať na rôzne veľkosti obrazoviek a orientácie zariadení. To je kľúčové pre moderné mobilné aplikácie, ktoré musia fungovať naprieč rôznymi typmi a veľkosťami zariadení.



Obrázok 4 Ukážka Dizajnu aplikácie

8.2 MainActivity.java

Súbor MainActivity.java predstavuje hlavnú časť užívateľského rozhrania v aplikácii a riadi interakcie užívateľa s aplikáciou. Dedí základné funkcie z triedy AppCompatActivity, ktorá poskytuje podporu pre rôzne verzie Androidu.

MainActivity začína svoju prácu v metóde onCreate, kde sa nastavuje vzhľad aplikácie podľa definície v súbore XML a inicializujú sa komponenty ako tlačidlá a textové polia.

```
@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
    accelerometerSensor = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    magnetometerSensor = sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);

    if (accelerometerSensor == null || magnetometerSensor == null)
    {
        Toast.makeText(context, this, text: "Accelerometer or Magnetometer not available", Toast.LENGTH_SHORT).show();
        finish();
        return;
    }

    tiltXTextView = findViewById(R.id.tilt_x);
    tiltYTextView = findViewById(R.id.tilt_y);

    Button startButton = findViewById(R.id.start_button);
    Button stopButton = findViewById(R.id.stop_button);
    Button resetButton = findViewById(R.id.reset_button);

    startButton.setOnClickListener(v -> startTracking());
    stopButton.setOnClickListener(v -> stopTracking());
    resetButton.setOnClickListener(v -> resetTracking());
}
```

Metóda onCreate v súbore MainActivity.java je základom pre každú Android aktivitu. Pri spustení aktivity, táto metóda nastavuje vizuálne rozhranie aplikácie pomocou príkazu setContentView(R.layout.activity_main), ktorý nahráva layout activity_main.xml. Tento layout obsahuje všetky užívateľské prvky rozhrania.

SensorManager slúži na správu senzorov zariadenia, konkrétne accelerometerSensor na meranie pohybu a magnetometerSensor na detekciu magnetických polí. Ak by tieto senzory neboli dostupné na zariadení, aplikácia by informovala užívateľa a ukončila by sa metódou finish(), aby sa predišlo chybám absencie týchto senzorov.

tiltXTextView a tiltYTextView sú komponenty, ktoré zobrazujú údaje o náklone zariadenia v osiach X a Y. Tieto dáta umožňujú užívateľovi vidieť ako sa zariadenie nakláňa v reálnom čase.

Aplikácia obsahuje 3 tlačidlá a to startButton, stopButton a resetButton. Každé tlačidlo má počúva udalosti v prípade stlačenie tlačidla. Tieto tlačidlá umožňujú užívateľovi začať sledovanie uhlov náklonu, zastaviť sledovanie, alebo resetovať.

```
private static class ClientTask extends AsyncTask<Float, Void, Void>
{
    1 usage
    private static final String IP_ADDRESS = "192.168.1.62";
    1 usage
    private static final int PORT = 55555;

    @Override
    protected Void doInBackground(Float... angles)
    {
        float tiltX = angles[0];
        float tiltY = angles[1];

        int tiltXX = (int) tiltX;
        int tiltYY = (int) tiltY;

        try (Socket socket = new Socket(IP_ADDRESS, PORT);
            PrintWriter out = new PrintWriter(socket.getOutputStream(), autoFlush: true))
        {
            out.println(tiltXX + " " + tiltYY);
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
        return null;
    }
}
```

Triedu ClientTask rozširuje trieda AsyncTask, ktorá slúži na spracovanie operácií na pozadí bez toho, aby bolo ovplyvnené užívateľské rozhranie. Účelom tejto triedy je zaslať údaje o náklone zariadenia na server prostredníctvom sieťového pripojenia.

Trieda ClientTask definuje dve konštanty: IP_ADDRESS a PORT, ktoré určujú sieťovú adresu a port servera, na ktorý sa majú údaje posielat'. V tomto projekte je IP adresa nastavená na 192.168.1.62 s portom 55555. Tieto konštanty sa používajú na vytvorenie socketového pripojenia k serveru.

Metóda doInBackground(Float... angles) je zavolaná vždy, keď je úloha spustená. Ako vstupné parametre prijíma polohu zariadenia, konkrétne hodnoty náklonu v osiach X a Y. Tieto hodnoty sú preformátované na celé čísla (int), aby posielali čísla bez desatinných miest.

V ClientTask sa vytvorí socket pomocou IP adresy a portu. PrintWriter automaticky odošle údaje o náklone oddelené medzerou bez zdržania vďaka flushu.

Na koniec metóda doInBackground vracia null, že úloha na pozadí bola dokončená.

```
@Override
protected void onResume()
{
    super.onResume();
    sensorManager.registerListener(listener: this, accelerometerSensor, samplingPeriodUs: 3);
    sensorManager.registerListener(listener: this, magnetometerSensor, samplingPeriodUs: 3);
}

@Override
protected void onPause()
{
    super.onPause();
    sensorManager.unregisterListener(this);
}
```

Metóda onResume sa zavolá keď sa aplikácia spustí. Aplikácia začne prijímať aktualizácie z akcelerometra a magnetometra cez sensorManager.

Metóda onPause sa zavolá keď napríklad užívateľ vypne aplikáciu, prejde na inú alebo keď zamkne obrazovku. Zavolá sa sensorManager.unregisterListener(this), čo odstráni poslucháče senzorov.

```
@Override
public void onSensorChanged(SensorEvent event)
{
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER)
    {
        accelerometerValues[0] = event.values[0];
        accelerometerValues[1] = event.values[1];
        accelerometerValues[2] = event.values[2];
    }
    else if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD)
    {
        magnetometerValues[0] = event.values[0];
        magnetometerValues[1] = event.values[1];
        magnetometerValues[2] = event.values[2];
    }

    if (isTracking)
    {
        updateOrientation();
    }
}
```

Metóda `onSensorChanged` sa aktivuje, keď dojde k zmene hodnôt zariadenia z akcelerometra alebo magnetometra.

Keď senzor zaznamená zmenu, aplikácia zavolá metódu `onSensorChanged` s objektom `SensorEvent`, ktorý obsahuje údaje zo senzora.

Ak senzor, ktorý vyvolal udalosť je akcelerometer (zistuje sa pomocou `event.sensor.getType() == Sensor.TYPE_ACCELEROMETER`), hodnoty "x", "y", "z" sa uložia do poľa `accelerometerValues`. Tieto hodnoty reprezentujú zrýchlenie zariadenia v týchto troch smeroch.

Ak senzor ktorý vyvolal udalosť je magnetometer (`event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD`), metóda uloží jeho hodnoty do poľa `magnetometerValues`. Tieto hodnoty predstavujú silu a smer magnetického poľa okolo zariadenia v osiach x, "y", "z".

Ďalej metóda kontroluje či je `isTracking` nastavené na `true` ak hej, tak sa zavolá metóda `updateOrientation`, ktorá spracováva získané údaje zo senzorov a aktualizuje uhol naklonenia.

```
private void updateOrientation()
{
    float[] rotationMatrix = new float[9];
    float[] orientationAngles = new float[3];
    SensorManager.getRotationMatrix(rotationMatrix, null, accelerometerValues, magnetometerValues);
    SensorManager.getOrientation(rotationMatrix, orientationAngles);

    float tiltX = (float) Math.toDegrees(orientationAngles[1]);
    float tiltY = (float) Math.toDegrees(orientationAngles[2]);

    if (tiltY > 90) { tiltY = 180 - tiltY; }
    else if (tiltY < -90) { tiltY = -180 - tiltY; }

    tiltX = (float) Math.round(tiltX);
    tiltY = (float) Math.round(tiltY);

    tiltXTextView.setText("Tilt X: " + (int)tiltX);
    tiltYTextView.setText("Tilt Y: " + (int)tiltY);

    tiltX = 2*tiltX;
    if(tiltX>90) {tiltX=90;}
    else if(tiltX<-90){tiltX=-90;}
    tiltY = 2*tiltY;
    if(tiltY>90) {tiltY = 90;}
    else if(tiltY<-90){tiltY=-90;}

    if (isTracking)
    {
        new ClientTask().execute(tiltX, tiltY);
    }
}
```

Metóda `updateOrientation` vypočíta orientáciu zariadenia pomocou údajov z akcelerometra a magnetometra. Tento proces sa rozdeľuje na niekoľko krokov:

- **Výpočet orientácie:** Najskôr sa použijú údaje zo senzorov na vytvorenie rotačnej matice ktorá určuje, ako je zariadenie naklonené a potom sa tieto údaje prevedú na orientačné uhly v stupňoch.
- **Korekcia tiltY:** Hodnoty Uhlu tiltY môžu presiahnuť +/-90 stupňov, preto pre zlepšenie sú použité dve podmienky, ktoré to jednoduchým výpočtom optimalizujú.
- **Aktualizácia displeja:** Upravené hodnoty náklonu sa zobrazia na obrazovke.
- **Odoslanie údajov:** Ak je aplikácia beží (`isTracking` je true), tak sa upravené údaje sa posielajú na server, ktorý je na Raspberry Pi.

Pri testovaní a ladení aplikácie sa ukázalo, že nutnosť nakloniť mobil o 90 stupňov na dosiahnutie maximálneho náklonu roviny je veľmi nepraktické. Preto som implementoval podmienky, ktoré zdvojnásobujú uhly naklonenia v oboch osiach a zároveň zabezpečujú, aby uhly neprekročili hodnoty 90 a -90 stupňov. Okrem toho aktualizujem zobrazenie uhlov na obrazovke ešte pred týmito podmienkami, aby uhly naklonenia zariadenia boli zobrazené správne.

```
private void startTracking()
{
    isTracking = true;
    updateOrientation();
}

1 usage
private void stopTracking()
{
    isTracking = false;
}

1 usage
private void resetTracking()
{
    tiltXTextView.setText("Tilt X: 0");
    tiltYTextView.setText("Tilt Y: 0");

    new ClientTask().execute(...params: 0f, 0f);
}
```

- **StartTracking():** Táto metóda aktivuje sledovanie orientácie zariadenia. Nastaví premennú `isTracking` na `true`. Po nastavení tejto premennej volá metódu `updateOrientation()`, ktorá ihneď aktualizuje zobrazenie orientácie zariadenia.
- **StopTracking():** Opačom predchádzajúcej metódy, `stopTracking()` zastaví sledovanie orientácie zariadenia tým, že nastaví `isTracking` na `false`. Tým sa efektívne deaktivuje aktualizácia orientácie.
- **ResetTracking():** `resetTracking()` slúži na resetovanie zobrazených hodnôt náklonu na nulu a odoslanie týchto nulových hodnôt na server prostredníctvom `ClientTask`. Toto je užitočné na vynulovanie stavu keď užívateľ chce začať hrať z pôvodného, neutrálneho stavu. Textové polia sa aktualizujú na "Tilt X: 0" a "Tilt Y: 0".

9 SERVER

```
import socket
import subprocess
from time import sleep
from gpiozero import AngularServo

servoX = AngularServo(18, min_pulse_width=0.0005, max_pulse_width=0.0025)
servoY = AngularServo(23, min_pulse_width=0.0005, max_pulse_width=0.0025)

def is_port_available(port):
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
        try:
            s.bind(('192.168.1.62', port))
            return True
        except OSError:
            return False

def start_server(port):
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as server_socket:
        server_socket.bind(('192.168.1.62', port))
        server_socket.listen()
        while True:
            client_socket, addr = server_socket.accept()
            handle_connection(client_socket)
```

Program zahrňuje importy knižníc potrebných na sieťovú komunikáciu, časovú synchronizáciu a ovládanie servo motorov.

servoX a servoY sú objekty, ktoré predstavujú servomotory pripojené k GPIO pinom 18 a 23 na Raspberry Pi. Parametre `min_pulse_width` a `max_pulse_width` určujú rozsah pulzov ktoré môže servo prijať.

Funkcia `is_port_available(port)` slúži na overenie, či je port dostupný na spustenie servera. Táto funkcia navezuje socket na daný port a IP adresu (192.168.1.62). Ak je port dostupný vráti True.

Funkcia `start_server(port)` vytvorí socket a naviaže ho na zadanú IP adresu a port. Tu server začne počúvať(`listen`) prichádzajúce spojenia.

Počúvanie: Metóda `listen()` nastaví server do režimu počúvania, kde čaká na prichádzajúce pripojenia.

V nekonečnej slučke server volá `accept()`, čo blokuje vykonávanie kódu, až kým sa nevytvorí nové pripojenie od klienta. Metóda `accept()` vráti nový socket (`client_socket`) a adresu klienta (`addr`), ktoré sú použité na komunikáciu.

Po prijatí spojenia sa volá funkcia `handle_connection`, ktorá spracuje prijaté dáta od klienta pomocou nového socketu.

```
def handle_connection(client_socket):
    with client_socket:
        while True:
            data = client_socket.recv(32)
            if not data:
                break
            #print(data.decode('utf-8').strip())
            servoX.angle = int(data.decode('utf-8').strip().split()[0])
            servoY.angle = int(data.decode('utf-8').strip().split()[1])

if __name__ == "__main__":
    port = 55555
    while True:
        if is_port_available(port):
            start_server(port)
            break
        else:
            sleep(1)
```

Funkcia `handle_connection(client_socket)` je zodpovedná za spracovanie komunikácie s klientom prostredníctvom socketu (`client_socket`).

Funkcia vstupuje do nekonečného cyklu, kde počúva prichádzajúce dáta od klienta. Na prijatie dát používa metódu `recv(32)`

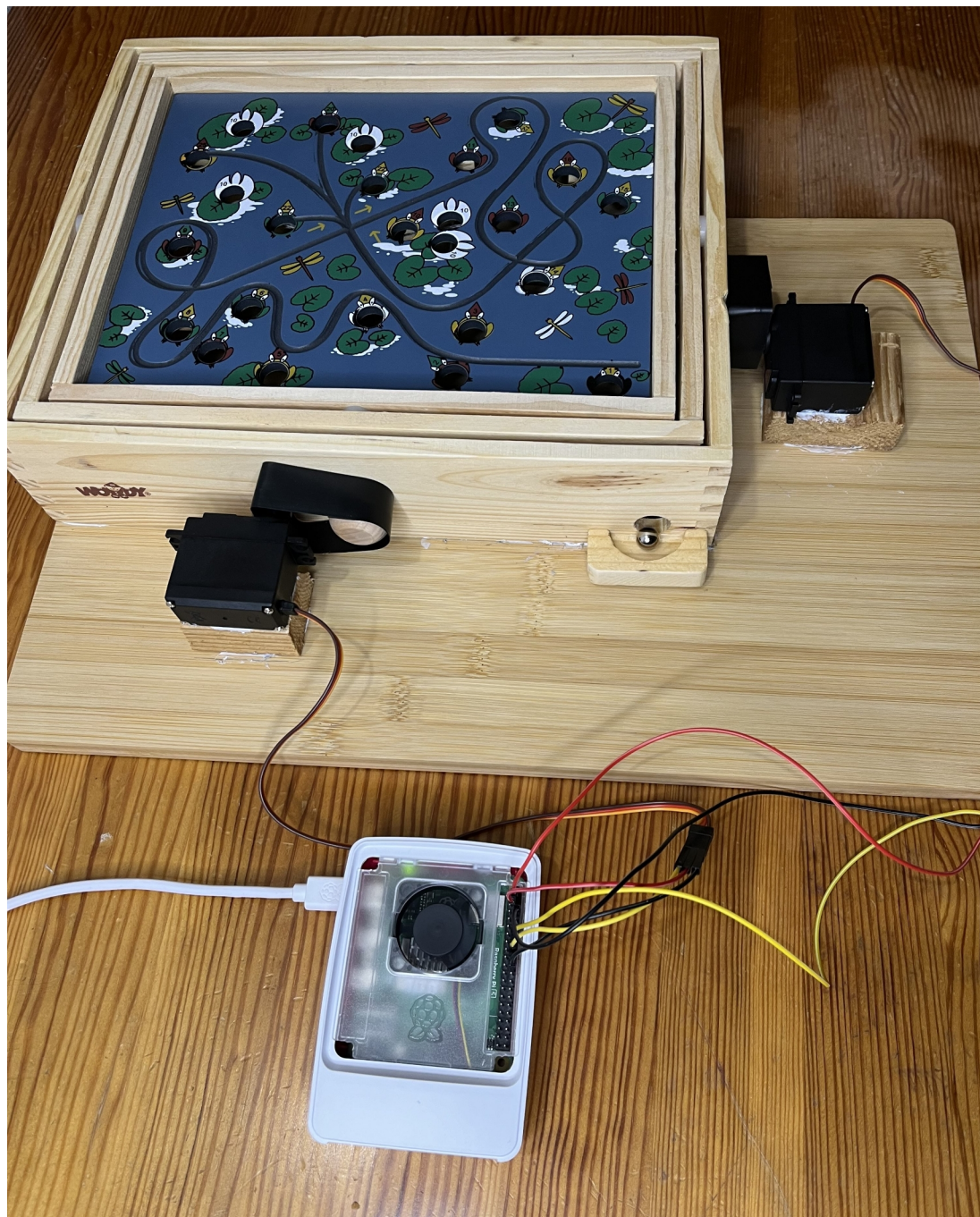
Ak metóda `recv` vráti prázdny reťazec "", znamená to, že klient zatvoril pripojenie. V takom prípade funkcia preruší cyklus pomocou príkazu `break`.

Ak sú prijaté nejaké dáta, funkcia ich dekóduje z formátu UTF-8 na reťazec, cez `split` rozdelí reťazec na dve časti. Tieto časti sú celé čísla, ktoré reprezentujú uhly pre servomotory (`servoX` a `servoY`).

Port, na ktorom server počúva je nastavený na 55555 a program neustále kontroluje, či je port dostupný na spustenie servera pomocou funkcie `is_port_available`. Ak je port dostupný, volá sa `start_server` s týmto portom a server začne fungovať. Ak nie program počká jednu sekundu a pokúsi sa o to znova.

10 HOTOVÝ PROJEKT

V praktickej časti tejto bakalárskej práce prezentujem realizáciu diaľkovo ovládanej naklonenej roviny, ktorú som prilepil na veľkú dosku. Táto doska slúži ako pevný a stabilný základ pre upevnenie všetkých komponentov, drevených kvádrov a servomotorov.



Obrázok 5 Finálna verzia projektu

10.1 Konštrukcia

Projekt sa skladá z dvoch hlavných servomotorov, ktoré sú pripevnené na dva malé drevené kvádre. Tieto kvádre slúžia ako podstava pre servomotory. Servomotory sú riadené cez Raspberry Pi, ktoré prijíma signály z mobilnej aplikácie. Táto aplikácia využíva senzory v mobilnom telefóne na zisťovanie náklonu zariadenia, čím ovláda náklon roviny.



Obrázok 6 Pripevnené servo

10.2 Funkčnost

System pracuje na princípe čítania údajov z akcelerometra a magnetometra v mobilnom telefóne. Tieto údaje sú spracované aplikáciou, ktorá ich prevedie na príkazy pre servomotory. Pri zmenách náklonu telefónu server na Raspberry Pi dynamicky upravuje uhol naklonenej roviny prostredníctvom servomotorov. Komunikácia medzi mobilným telefónom a Raspberry Pi prebieha cez sieť. Tento projekt demonštruje využitie moderných technológií a embedded systémov pre realizáciu praktických a interaktívnych aplikácií. Je príkladom toho, ako možno jednoduché komponenty spojiť do funkčného celku.

11 POROVNANIE ALTERNATÍVNEJ TECHNOLOGIE

Článok "How to Control a Servo Motor Using Arduino"[32] poskytuje alternatívne riešenie pre ovládanie servomotorov v rámci embedded systémov, pričom sa zameriava na platformu Arduino. Arduino je ideálne pre projekty, kde je potrebné rýchle prototypovanie a kde je dôležitá priama interakcia s hardvérom. V článku sú detailne popísané základy ovládania servomotorov, ktoré sú nevyhnutné pre aplikácie v robotike a automatizácii.

Výhodou Arduina je jeho schopnosť poskytnúť nízkonákladové a efektívne riešenie pre školské projekty a hobby aplikácie.

Naproti tomu, môj projekt, ktorý využíva Raspberry Pi na ovládanie servomotorov cez sieť, ponúka sofistikovanejšie riešenie, ktoré integruje pokročilé sieťové funkcie pre diaľkové ovládanie. Raspberry Pi teda predstavuje robustnejšiu platformu pre implementáciu v náročnejších priemyselných a výskumných projektoch, kde je žiadúca vyššia výpočtová sila a flexibilita.

Oba prístupy majú svoje miesto v embedded systémov. Zatiaľ čo Arduino je výborné pre začiatočnicke projekty a vzdelávacie účely, Raspberry Pi ponúka rozšírené možnosti pre komplexnejšie aplikácie.

ZÁVĚR

Táto bakalárska práca sa venovala vývoju a implementácii systému na diaľkové ovládanie náklonu roviny, pričom kľúčovú úlohu v ňom zohrával chytrý telefón. Cieľom bolo vytvoriť plne funkčný systém, ktorý by umožňoval užívateľovi ovládať náklon roviny prostredníctvom náklonu svojho telefónu. Tento cieľ bol úspešne dosiahnutý prostredníctvom dôkladného výberu hardvérových komponentov, efektívneho programovania aplikácie a integrácie s mobilným zariadením, čo viedlo k vývoju robustného a spoľahlivého ovládacieho systému.

V priebehu vypracovania boli splnené všetky stanovené ciele. Bola vykonaná stručná literárna rešerš, ktorá poskytla dôležitý teoretický základ pre pochopenie embedded systémov. Boli popísané metódy programovania mobilných aplikácií pre Android aj iOS, vrátane práce so senzorickým systémom chytrého telefónu. Bola vybraná a modifikovaná komerčne dostupná hra s rovinou a boli navrhnuté vhodné akčné členy a riadiaca jednotka pre jeho automatizáciu. Systém bol úspešne zostavený a naprogramovaný, s riadiacou jednotkou schopnou prijímať žiadané hodnoty náklonu z nadriadeného systému. Aplikácia pre chytrý telefón bola vytvorená a integrovaná s riadiacim systémom, čo umožnilo dynamické ovládanie naklonenej roviny. Systém bol komplexne otestovaný a odladený, čo potvrdilo jeho funkčnosť a splňanie všetkých požiadaviek.

Výsledkom práce je teda systém, ktorý nielenže splňa všetky požiadavky ale zároveň tak prispieva k rozvoju technológií v oblasti diaľkového ovládania a otvára dvere pre ďalšie výskumy a inovácie.

SEZNAM POUŽITÉ LITERATURY

- [1] HEAVY.AI. Embedded Systems. Online. C2024. Dostupné z: <https://www.heavy.ai/technical-glossary/embedded-systems>. [cit. 2024-05-12].
- [2] ASHTARI, Hossein. What Are Embedded Systems? Meaning, Components, and Applications. Online. 2023. Dostupné z: <https://www.spiceworks.com/tech/tech-general/articles/what-are-embedded-systems/>. [cit. 2024-05-12].
- [3] EVANS, Henry. Main Types of Embedded Systems Worth Knowing. Online. 2023. Dostupné z: <https://www.velvtech.com/blog/types-of-embedded-systems/>. [cit. 2024-05-12].
- [4] GEEKSFORGEEKS. Difference between Hard real time and Soft real time system. Online. 2023. Dostupné z: <https://www.geeksforgeeks.org/difference-between-hard-real-time-and-soft-real-time-system/>. [cit. 2024-05-12].
- [5] LUTKEVICH, Ben. Embedded system. Online. 2020. Dostupné z: <https://www.techtarget.com/iotagenda/definition/embedded-system>. [cit. 2024-05-12].
- [6] BASUMALLICK, Chiradeep. What Is Raspberry Pi? Models, Features, and Uses. Online. 2022. Dostupné z: <https://www.spiceworks.com/tech/networking/articles/what-is-raspberry-pi/>. [cit. 2024-05-12].
- [7] RED HAT. What is a Raspberry Pi? Online. C2024. Dostupné z: <https://opensource.com/resources/raspberry-pi>. [cit. 2024-05-12].
- [8] MOMETRIX. What is a Raspberry Pi and How Does it Work? Online. C2024. Dostupné z: <https://www.piday.org/whats-a-raspberry-pi-and-how-does-it-work/>. [cit. 2024-05-12].
- [9] ELEKTOR. Raspberry Pi 5 (8 GB RAM) + FREE Raspberry Pi 5 Essentials (E-book). Online. C2024. Dostupné z: <https://www.elektor.com/products/raspberry-pi-5-8-gb-ram>. [cit. 2024-05-12].
- [10] LINKEDIN. How does ARM architecture compare to other architectures in terms of ease of use? Online. C2024. Dostupné z: https://www.linkedin-com.translate.google.com/advice/0/how-does-arm-architecture-compare-other-architectures-9gdvc?_x_tr_sl=en&_x_tr_tl=sk&_x_tr_hl=sk&_x_tr_pto=sc. [cit. 2024-05-12].
- [11] RASPBERRY PI. Raspberry Pi Documentation. Online. C2012-2024. Dostupné z: <https://www.raspberrypi.com/documentation/computers/processors.html>. [cit. 2024-05-12].
- [12] RASPBERRY PI. BCM2711 ARM Peripherals. Online. C2012-2024. Dostupné z: <https://datasheets.raspberrypi.com/bcm2711/bcm2711-peripherals.pdf>. [cit. 2024-05-12].

- [13] POUNDER, Les. Raspberry Pi GPIO Pinout: What Each Pin Does on Pi 4, Earlier Models. Online. 2023. Dostupné z: <https://www.tomshardware.com/reviews/raspberry-pi-gpio-pinout,6122.html>. [cit. 2024-05-12].
- [14] VERIZON. Wi-Fi. Online. 2023. Dostupné z: <https://www.verizon.com/articles/internet-essentials/wifi-definiton/>. [cit. 2024-05-12].
- [15] SELPH, Chris. 802.11 Standards Explained: 802.11ax, 802.11ac, 802.11b/g/n, 802.11a. Online. 2023. Dostupné z: <https://www.lifewire.com/wireless-standards-802-11a-802-11b-g-n-and-802-11ac-816553>. [cit. 2024-05-12].
- [16] AGUIRRE, Austin. Understanding Wi-Fi Frequencies: Choosing Between 2.4 GHz and 5 GHz. Online. 2022. Dostupné z: <https://www.highspeedinternet.com/resources/2-4-ghz-vs-5-ghz-wi-fi>. [cit. 2024-05-12].
- [17] FRANKLIN, Curt a POLLETTE, Chris. How Bluetooth Works. Online. C2024. Dostupné z: <https://electronics.howstuffworks.com/bluetooth.htm>. [cit. 2024-05-12].
- [18] What is Linux? Online. C2024. Dostupné z: <https://opensource.com/resources/linux>. [cit. 2024-05-12].
- [19] LONG, Moe. What is Raspberry Pi OS? Online. 2020. Dostupné z: <https://www.electromaker.io/blog/article/what-is-raspberry-pi-os>. [cit. 2024-05-12].
- [20] ONDARA, Winnie. Debian 12 "Bookworm" Release | What's New? Online. 2023. Dostupné z: <https://www.cherryservers.com/blog/debian-12-bookworm-release>. [cit. 2024-05-12].
- [21] IQS DIRECTORY. Electric Motor. Online. C2024. Dostupné z: <https://www.iqsdirectory.com/articles/electric-motor.html>. [cit. 2024-05-12].
- [22] KOHLI, Venus. Stepper vs Servo Motors: What's the Difference? Online. 2023. Dostupné z: <https://www.wevolver.com/article/stepper-vs-servo-motors-a-comprehensive-comparison-for-your-next-project>. [cit. 2024-05-12].
- [23] TAYLOR, Eliza. Android vs iOS: A Detailed Comparison. Online. 2024. Dostupné z: <https://www.theknowledgeacademy.com/blog/android-vs-ios/>. [cit. 2024-05-12].
- [24] DUTTA, Subhasish. Android vs. iOS App Development - A Quick Analysis. Online. C2024. Dostupné z: <https://www.turing.com/resources/mobile-app-development>. [cit. 2024-05-12].
- [25] Sensors Overview. Online. 2024. Dostupné z: https://developer.android.com/develop/sensors-and-location/sensors/sensors_overview. [cit. 2024-05-12].
- [26] GEEKSFORGEEKS. What is Android Studio? Online. 2024. Dostupné z: <https://www.geeksforgeeks.org/overview-of-android-studio/>. [cit. 2024-05-12].

- [27] AMAZON WEB SERVICES. What is Java? Online. C2024. Dostupné z: <https://aws.amazon.com/what-is/java/>. [cit. 2024-05-12].
- [28] MICROSOFT. What is Java? Online. C2024. Dostupné z: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-java-programming-language>. [cit. 2024-05-12].
- [29] SILVERIO, Manuel. What Is Kotlin? Online. 2022. Dostupné z: <https://builtin.com/software-engineering-perspectives/kotlin>. [cit. 2024-05-12].
- [30] STAFF, Coursera. What Is Python Used For? A Beginner's Guide. Online. 2024. Dostupné z: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>. [cit. 2024-05-12].
- [31] NUTTALL, Ben. Gpiozero 2.0.1 Documentation Release 2.0.1. Online. 2024. Dostupné z: <https://readthedocs.org/projects/gpiozero/downloads/pdf/stable/>. [cit. 2024-05-12].
- [32] AUTOMATIC ADDISON. How to Control a Servo Motor Using Arduino. Online. 2019. Dostupné z: <https://automaticaddison.com/how-to-control-a-servo-motor-using-arduino/>. [cit. 2024-05-12].

SEZNAM OBRÁZKŮ

Obrázok 1 Raspberry Pi 5 4GB RAM.....	13
Obrázok 2 40-Pinový header na Raspberry Pi[13].....	17
Obrázok 3 Príkaz na zistenie verzie OS.....	24
Obrázok 4 Ukážka Dizajnu aplikácie.....	38
Obrázok 5 Finálna verzia projektu.....	47
Obrázok 6 Pripevnené servo.....	48

SEZNAM PŘÍLOH

Příloha P 1: zoznam příloh na cd.

PŘÍLOHA P 1: ZOZNAM PŘÍLOH NA CD.

fulltext.pdf – text bakalárskej práce

prilohy.zip – archív obsahujúci všetky zdrojové kódy a demo.mov