

Sustav za praćenje kinematičkih parametara skijaša

Sabljić, Antonio

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:168:702468>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-28**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 350

**SUSTAV ZA PRAĆENJE KINEMATIČKIH PARAMETARA
SKIJAŠA**

Antonio Sabljic

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 350

**SUSTAV ZA PRAĆENJE KINEMATIČKIH PARAMETARA
SKIJAŠA**

Antonio Sabljic

Zagreb, lipanj 2024.

DIPLOMSKI ZADATAK br. 350

Pristupnik: **Antonio Sablić (0036516426)**

Studij: Računarstvo

Profil: Računalno inženjerstvo

Mentor: izv. prof. dr. sc. Igor Čavrak

Zadatak: **Sustav za praćenje kinematičkih parametara skijaša**

Opis zadatka:

Neprestano povećanje performanci i smanjenje dimenzija računalnih sustava i osjetila omogućava širenje njihove primjene u sve zahtjevnije okoline. Jedna od recentnijih primjena je u praćenju kinematičkih parametara skijaša, gdje takvi sustavi istovremeno prate veći broj parametara te, tijekom ili po završetku treninga, daju ciljane ili sveobuhvatne analize i savjete kako poboljšati tehniku skijanja. Cilj ovog diplomskog rada je osmisliti koncept i ostvariti osnovni sustav za praćenje kinematičkih parametara skijaša. Potrebno je proučiti mogućnosti na tržištu dostupnih sustava, te predložiti osnovnu arhitekturu sustava zasnovanu na tri sloja; sloju osjetila, posredničkom sloju i sloju usluga u računalnom oblaku. S obzirom na predloženu arhitekturu, potrebno je implementirati osnovne komponente sva tri sloja; u najnižem sloju potrebno je izvesti komponente za mjerenje odabranih kinematičkih parametara skijaša temeljene na mikrokontroleru ESP32 i pripadajućim osjetilima. Funkcionalnost posredničkog sloja, obradu i prosljeđivanje prikupljenih podataka te sučelje prema korisniku, izvesti na mobilnoj platformi, a za analize podataka i njihovu pohranu na strani poslužitelja treba koristiti jedan od sustava otvorenog kôda za upravljanje tokovima podataka. Na kraju, potrebno je izgraditi testnu okolinu za ispitivanje performanci sustava s naglaskom na procjenu točnosti analiza i brzinu odziva sustava.

Rok za predaju rada: 28. lipnja 2024.

Sadržaj

1. Uvod	1
2. Arhitektura sustava	2
2.1. Sloj osjetila i prikupljanje podataka	4
2.2. Mobilna aplikacija i obrada podataka	6
2.3. Oblak i agregacija podataka	8
2.4. Vizualizacija podataka	8
3. Algoritam za procjenu skijaške vještine i broja zavoja	11
4. Tehnologije sustava	14
4.1. ESP32-WROOM-32	14
4.2. MPU6050	16
4.2.1. Akcelerometar	16
4.2.2. Žiroskop	17
4.3. Flutter	17
4.4. Spring	18
4.5. Kafka	19
4.5.1. Prijenos događaja	19
4.5.2. Način rada	20
4.5.3. Glavni koncepti i terminologija	20
4.6. React	22
4.7. Bluetooth	23
4.7.1. Bluetooth Classic	23
4.7.2. Bluetooth Low Energy (LE)	23
4.8. HTTP	26

5. Testiranje sustava	28
5.1. Simulacija realističnog skijanja	29
5.2. Simulacija idealnog skijanja	31
5.3. Simulacija sistematično slabijeg naginjanja unutarnje skije	33
5.4. Simulacija kašnjenja rubljenja unutarnje skije u početku zavoja	34
5.5. Testiranje pozicija ekstrema	35
5.6. Simulacija jedne skije	37
5.7. Simulacija unakrsnog kretanja skija	38
6. Daljnji rad i razmatranje	41
7. Zaključak	43
Literatura	44
Sažetak	45
Abstract	46
A: The Code	47

1. Uvod

Skijaška ovisnost o korekciji skijaških vještina je inspirirala kreiranje sustava sa kojim se mogu analizirati parametri kako bi se ispravile vještine skijanja. Primarno se gleda kut nagiba skija, cilj je imati u svakom trenutku kut nagiba skija isti. Inspiracija ovom diplomskom radu je sustav Carv koji koristi uloške za pancerce i mobilnu aplikaciju za analiziranje parametra skijanja. Glavna tema rada je izrada sustava koji će u stvarnom vremenu prikazivati parametre skijaša.

Razvoj ovakvog sustava zahtijeva duboko razumijevanje biomehanike skijanja, kao i tehnoloških aspekata senzora i podataka u stvarnom vremenu. U skijanju, precizna kontrola i prilagodba nagiba skija ključna je za optimalnu izvedbu i sigurnost skijaša. Sustav koji može kontinuirano pratiti i analizirati te parametre može značajno doprinijeti poboljšanju skijaških vještina, smanjenju rizika od ozljeda i povećanju zadovoljstva skijaša.

Sustav Carv, koji koristi senzorske uloške i mobilnu aplikaciju, postavlja standard u ovoj oblasti. Cilj je razviti sličan sustav koji će omogućiti analizu skijaških vještina u stvarnom vremenu. Korištenjem naprednih senzora i algoritama za obradu podataka, sustav će moći pružiti trenutačne povratne informacije skijašu, omogućujući mu da odmah ispravi eventualne pogreške i prilagodi svoju tehniku.

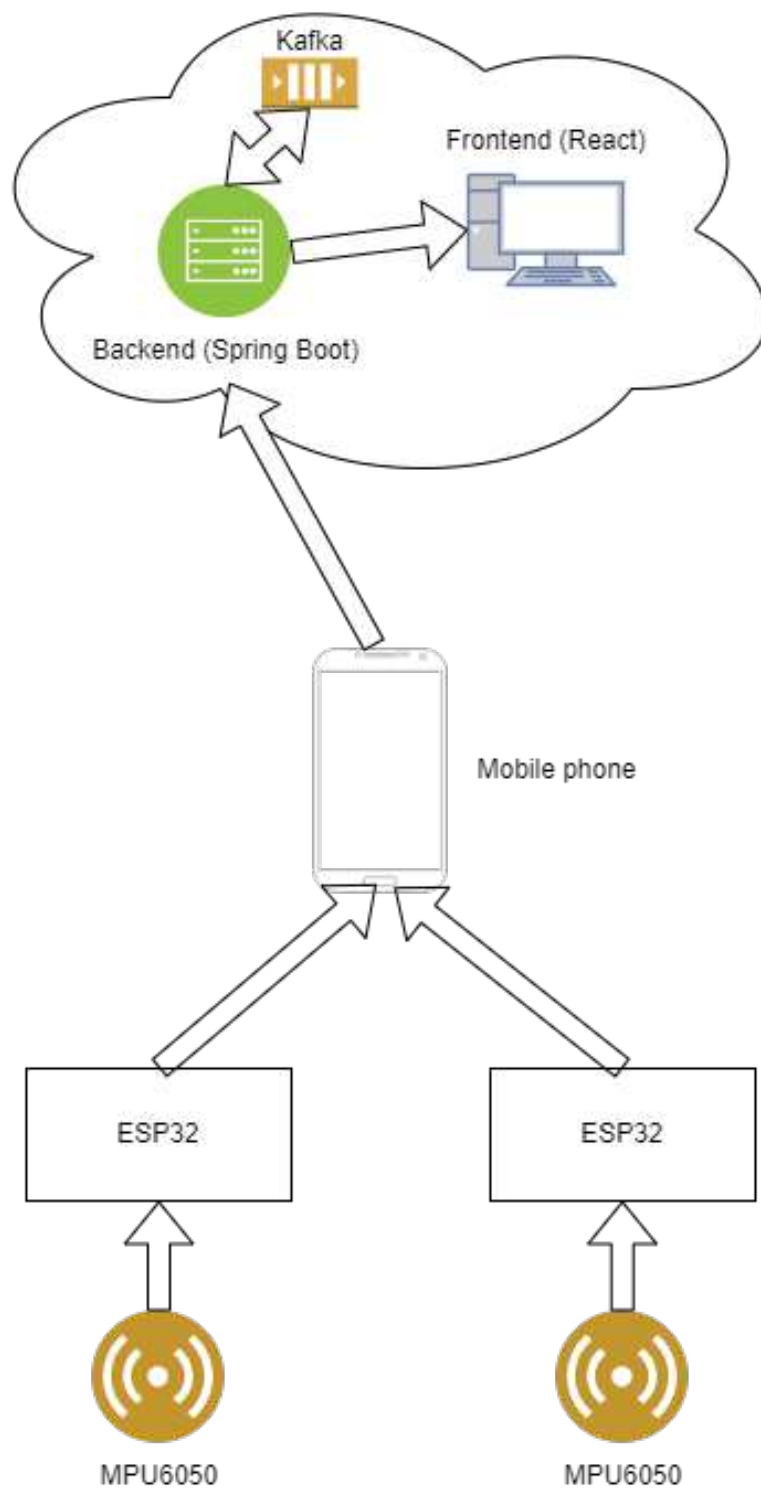
2. Arhitektura sustava

Sustav se sastoji od dva ESP32 mikrokontrolera koji uzimaju uzorke nagiba sa dva MPU6050 senzora u određenom vremenskom intervalu zadanom na mikrokontroleru i obavješćuju mobilni uređaj preko Bluetooth LE veze koje su vrijednosti nagiba skija u određenom vremenu, mobilni uređaj tokom vremena prikazuje trenutnu vrijednost nagiba oba senzora.

Nakon prikupljanja podataka, mobilni uređaj periodično šalje podatke u tijelu POST HTTP zahtjeva na server. Ova komunikacija između mobilnog uređaja i servera omogućava centralizirano prikupljanje i obradu podataka, što je ključno za daljnju analizu i vizualizaciju. Server zatim producira poruku za Kafku, distribuirani sustav za prijenos podataka, koja sprema poruku u temu (engl. *topic*) pod nazivom *measurements*

Nakon što poslužiteljska strana detektira poruku u temi *measurements*, konzumira je i šalje podatke na klijentsku stranu putem WebSocket veze. WebSocket veza omogućava dvosmjernu komunikaciju u stvarnom vremenu između servera i klijentske aplikacije, osiguravajući da se podaci odmah prenose i prikazuju korisnicima. Veza se nalazi na adresi **`http://localhost:8080/ws-message`** (veza se nalazi na IP adresi oblaka ali radi lakše interpretacije stavljeno je da je ta adresa localhost), na `/topic/message`.

Klijentska strana zatim reformatira dobivene podatke i prikazuje ih grafički. Korištenjem napredne JavaScript biblioteke za vizualizaciju Recharts [1] za crtanje grafova, korisnici mogu lako interpretirati podatke. Grafovi omogućuju jasno i pregledno očitavanje mjerenja nagiba, što pomaže skijašima da razumiju svoje tehnike i identificiraju područja za poboljšanje.



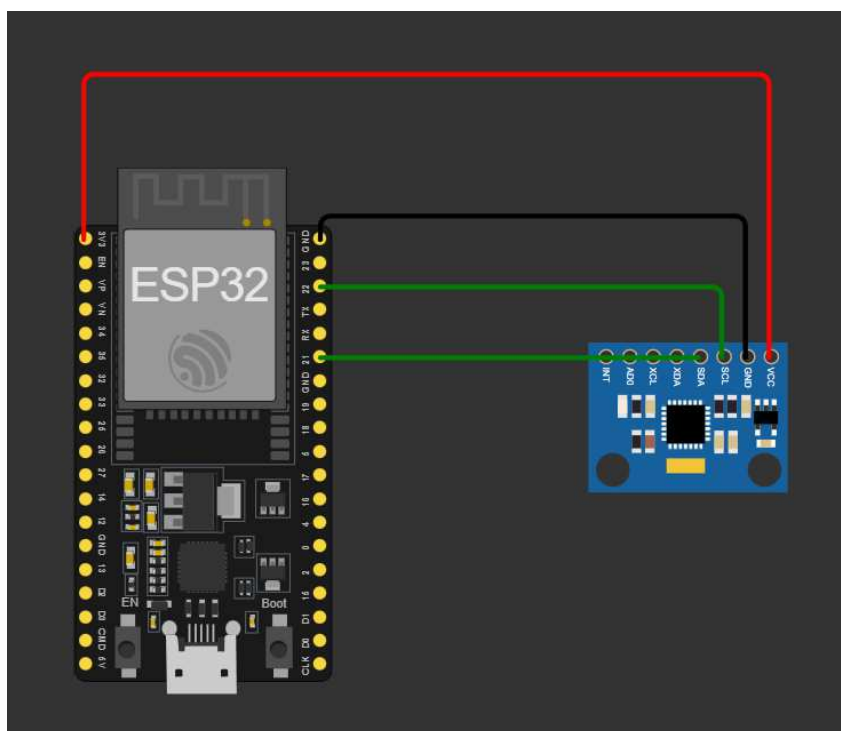
Slika 2.1. Izvedba sustava

2.1. Sloj osjetila i prikupljanje podataka

Sloj osjetila se sastoji od dva ESP32 mikrokontrolera i dva MPU6050 senzora koji su spojeni I2C komunikacijom.

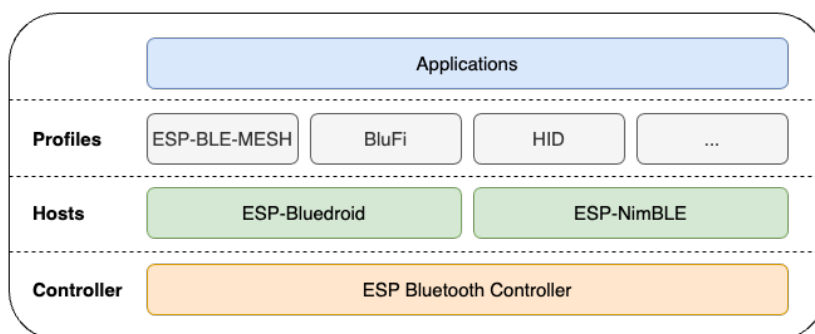
I2C je serijski, sinkroni, polu-dupleksni komunikacijski protokol za više uređaja koji omogućuje koegzistenciju više glavnih (engl. *master*) i podređenih (engl. *slave*) uređaja na istoj sabirnici. I2C koristi dvije dvosmjerne linije otvorenog odvoda (engl. *open-drain*) serijska linija podataka (SDA) i serijska linija sata (SCL). ESP32 ima 2 I2C kontrolera odgovorna za upravljanje komunikacijom na I2C sabirnici. I2C kontroler može biti glavni ili podređeni. Tipično, I2C podređeni uređaj ima 7-bitnu ili 10-bitnu adresu. ESP32 podržava I2C standardni način (engl. *Standard-mode*) i brzi način rada (engl. *Fast-mode*) koji mogu doseći frekvenciju do 100KHz odnosno 400KHz.[2]

Slika 2.2. prikazuje ESP32 koji je žično povezan na MPU6050 senzor gdje se koriste SDA (engl. *Serial Data*) i SCL (engl. *Serial Clock*) pin za prijenos podataka i nožice za napajanje senzora.



Slika 2.2. Arhitektura sloja osjetila

ESP32 koristi Bluetooth LE način rada za Bluetooth povezivanje 2.3.

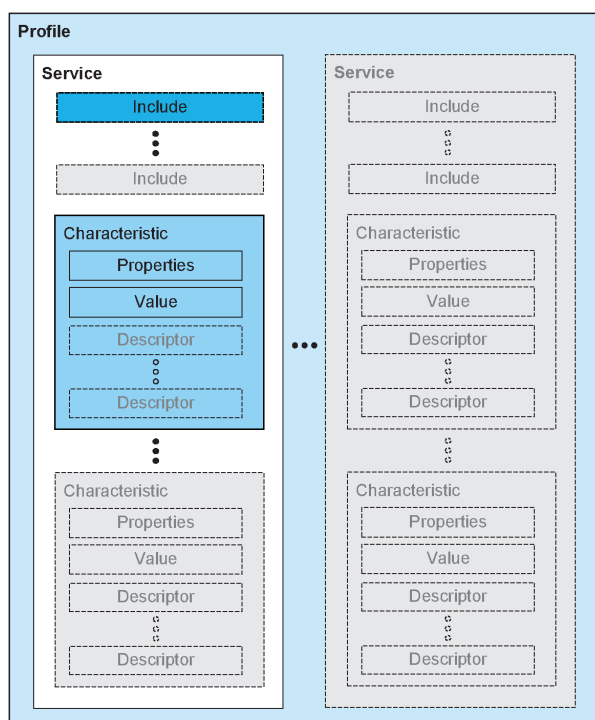


Slika 2.3. Arhitektura Bluetooth stoga ESP32

ESP-Bluetooth je modificirana verzija izvornog Android Bluetooth stoga. Bluetooth se sastoji se od dva sloja: Bluetooth gornjeg sloja (BTU) i sloja Bluetooth prijenosnog kontrolera (BTC). BTU sloj je odgovoran za obradu Bluetooth protokola donjeg sloja kao što su L2CAP, GATT/ATT, SMP, GAP i drugi profili. BTU sloj pruža sučelje s prefiksom "bta". BTC sloj je uglavnom odgovoran za pružanje sučelja s prefiksom "esp" aplikacijskom sloju, obradu profila temeljenih na GATT-u i rukovanje raznim zadacima.[3]

ESP-Bluetooth za ESP32 podržava Classic Bluetooth i Bluetooth LE.

Pri spajanju ESP32 mikrokontrolera na napajanje, ESP32 ulazi u fazu oglašivanja i čeka da se mobilni uređaj pretplati na karakteristiku 0xFF01 koja se nalazi na servisu 0x00FF.



Slika 2.4. Prikaz hijerarhije GATT profila [4]

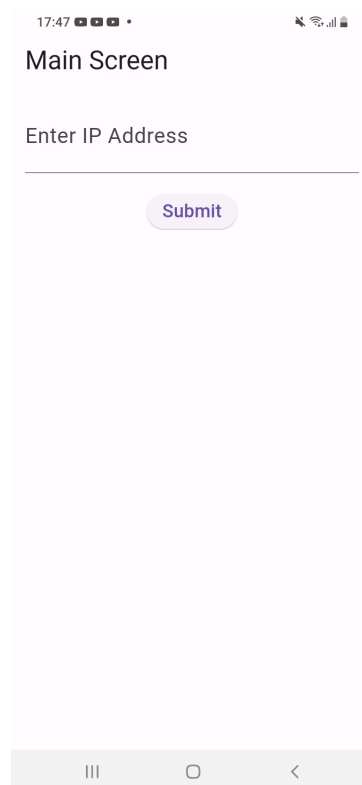
Na karakteristici 0xFF01 se zapisuju podatci sa senzora u obliku polja [x,y].

- x - označava smjer nagiba skije, može biti 0 ili 1
- y - označava nagib kuta, nagib se kreće od 0x00(0) do 0x64(100) gdje se postiže vrijednost 0x64 kada je senzor nagnut 90°, a 0x00 kada je nagnut 0°

2.2. Mobilna aplikacija i obrada podataka

Mobilna aplikacija sustava služi kao međusloj između oblaka i sloja osjetila. Naime, u mobilnoj aplikaciji se podatci sa sloja osjetila prikupljaju dok se ne skupi 25 mjerenja koji se reformatiraju i onda se šalje POST zahtjev na adresu oblaka **<http://localhost:8080/publish>**.

Adresa oblaka se upisuje na početnoj stranici aplikacije 2.5.



Slika 2.5. Početni zaslon na mobilnoj aplikaciji

Podatci sa senzora u obliku [x,y] se konvertiraju u JSON format i stavljaju u tijelo POST zahtjeva:

```
1 {  
2   "timestamp": "2024-04-23T18:40:50",
```

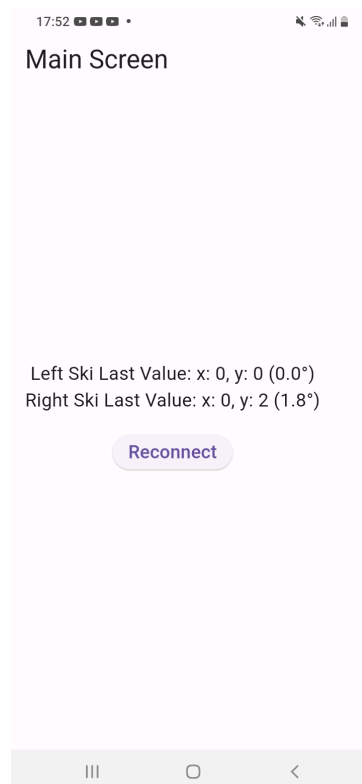
```

3  "measurements": {
4      "left_ski": [30, 35, 36, 50, 80],
5      "right_ski": [32, 38, 30, 30, 78]
6  },
7  "measurement_delay": 500
8  }

```

- timestamp - označava kada se uzeo prvi uzorak (u ovom slučaju je 30 i 32).
- measurements - označava mjerenja sa senzora u stupnjevima, gleda se u parovima npr. 30 i 32, 35 i 38, ...
- measurement delay - označava razmak između očitavanja u mjerenjima u milisekundama

Poslije spajanja na oblak na mobilnoj aplikaciji je prikazan trenutni nagib skija 2.6. Prikazan nagib je reprezentiran u dva oblika, u obliku dobiven sa ESP-a i u stupnjevima.

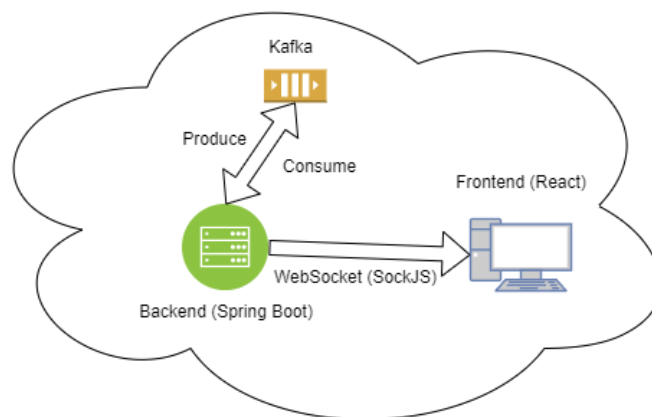


Slika 2.6. Zaslona sa trenutnim nagibom kuta skija

2.3. Oblak i agregacija podataka

Oblak se sastoji od tri dijela:

- Poslužiteljska strana (engl. *Backend*) - prima HTTP zahtjeve sa mobilnog uređaja
- Klijentska strana (engl. *Frontend*) - prikazuje vrijednosti dobivene od poslužiteljske strane
- Kafka - sprema dobivene podatke sa poslužiteljske strane u temu *measurements*



Slika 2.7. Arhitektura oblaka

Podatci u JSON formatu dolaze u tijelu HTTP POST zahtjeva na adresu **<http://localhost:8080/publish>**, nakon toga se podatci produciraju za Kafka temu *measurements*.

Nakon toga poslužiteljska strana uočava promjene na Kafka temi *measurements* i konzumira podatke te ih šalje na WebSocket točku **<http://localhost:8080/ws-message>** sa krajnjom točkom */topic/message*

2.4. Vizualizacija podataka

Vizualizacija podataka se odvija na klijentskoj strani u oblaku gdje klijentska strana sluša promjene na **<http://localhost:8080/ws-message>**, krajnjoj točki */topic/message*. Za prikaz grafova korištena je JavaScript biblioteka [Recharts](#)[1].

Kod linijskog grafa 2.8. zelena linija predstavlja lijevu skiju, a ljubičasta linija desnu

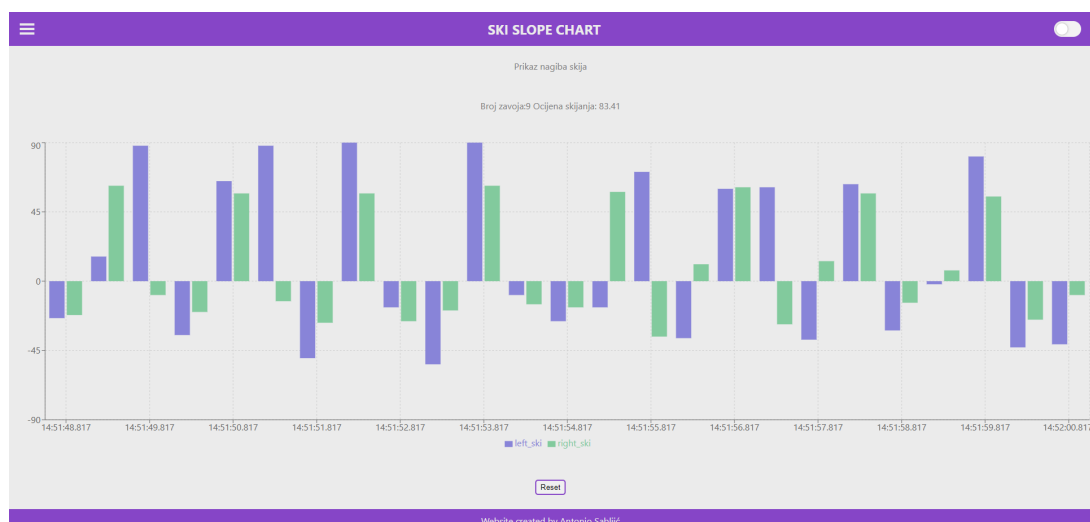
skiju.



Slika 2.8. Linijski graf

Na x-osi se nalaze vremenske oznake sa vremenom kada je napravljeno uzorkovanje. Na y-osi se nalaze podatci o nagibu skija u stupnjevima u određenom vremenu. U svakom trenutku pokazivačem se može očitati na grafu kakav je nagib skija.

Kod stupičastog grafa 2.9. situacija je slična, umjesto linija koji prikazuju nagib su stupci. Stupičasti graf 2.9. je malo pogodniji za preciznije i lakše očitavanje kuta nagiba skija dok je linijski graf 2.8. pogodniji za generalni pregled simetrije skija u vremenu.



Slika 2.9. Stupičasti graf

Kao i na linijskom grafu x-os predstavlja vrijeme i sadrži vremenske oznake sa vremenom kada je napravljeno uzorkovanje. Na y-osi su podatci o nagibima skija u određenom

vremenu i u svakom trenutku možemo pokazivačem pokazati kakvo je stanje skija u određenom trenutku.

Također u oba grafa postoji gumb **Reset** kojim možemo graf izbrisati.

Algoritam računanja broja zavoja i vještine skijanja 3. također daje dodatne parametre koji se nalaze na vrhu grafa.

3. Algoritam za procjenu skijaške vještine i broja zavoja

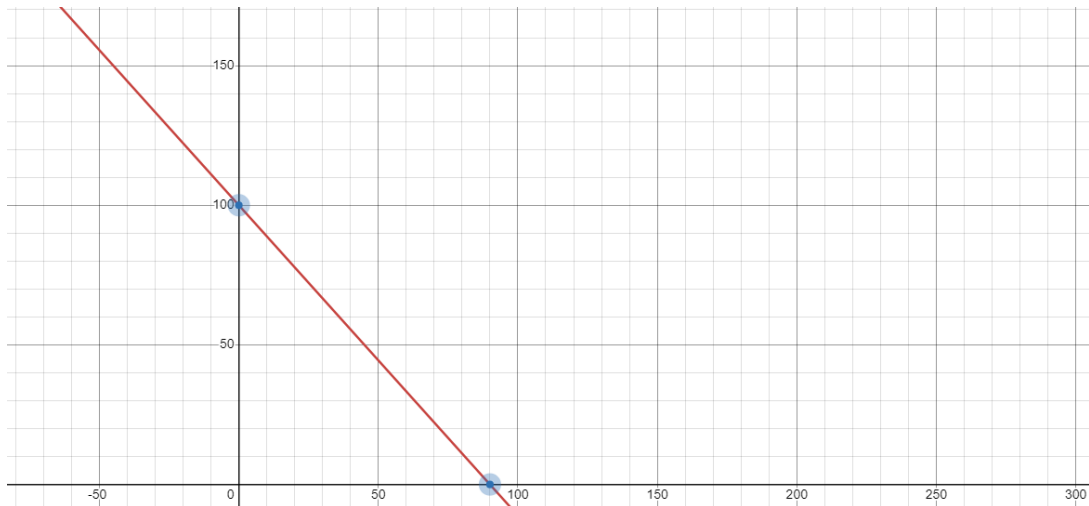
Kada se dobije set podataka sa kojim se prikazuje graf računa se broj zavoja i ocjena skijaške vještine.

Algoritam radi na principu da uzima parove mjerenja sa lijeve i desne skije i provjerava njihov nagib. Sve dok kod parova mjerenja postoji nagib obje skije isključivo pozitivan (sa uključenom nulom) ili isključivo negativan (sa uključenom nulom) uzima se kao jedan zavoj. Ako je jedna skija u pozitivnom smjeru a druga u negativnom zanemaruje se mjerenje i gledaju se ostali parovi mjerenja.

Nakon zaključivanja koji parovi predstavljaju zavoje izvodimo operaciju apsolutne razlike parova mjerenja i spremamo te vrijednosti u polje. Cilj skijaša je imati što paralelnije skije u svakom trenutku zavoja tako da što je razlika između skija manja to je rezultat bolji.

Svaku vrijednost iz polja vrijednosti apsolutne razlike parova mjerenja provodimo kroz funkciju:

$$f(x) = \begin{cases} \text{undefined} & \text{if } x < 0 \\ y = -\frac{10}{9}x + 100 & \text{if } 0 \leq x \leq 90 \\ \text{undefined} & \text{if } x > 90 \end{cases}$$



Slika 3.1. Graf funkcije ocjene skijanja

Prema 3.1. najveći kut razlike je 90° i to je kada su skije okomite jedna na drugu i tada je ocjena skijanja 0. U suprotnom ako su skije paralelne onda je ocjena 100.

Nakon dobivanja svih ocjena radi se izračun prosjeka i to je konačna ocjena skijaške vještine.

Algoritam:

```

1
2 function getSkiGrade(array) {
3   const gradeField = array.map((value)=>{
4     if(value >= 90) {
5       return 0;
6     }
7     return -1.11*value + 100;
8   });
9   return getAverage(gradeField);
10 }
11
12 const calculateCurveSimilarityAndNumberOfTurns = (data) => {
13   const { timestamp, measurements, measurement_delay } = data;
14   const { left_ski, right_ski } = measurements;
15   let curvN = 0;
16   let curvFieldAvg = [];

```

```

17 let k = 0;
18 while(k < left_ski.length) {
19     if(left_ski[k]>=0 && right_ski[k]>=0) {
20         let temp = [];
21         ++curvN;
22         let j = k;
23         while(left_ski[j]>=0 && right_ski[j]>=0) {
24             temp.push(Math.abs(left_ski[j] - right_ski[j]));
25             ++j;
26             k=j;
27         }
28         curvFieldAvg.push(getAverage(temp));
29     } else if (left_ski[k]<=0 && right_ski[k]<=0) {
30         let temp = [];
31         ++curvN;
32         let j = k;
33         while(left_ski[j]<=0 && right_ski[j]<=0) {
34             temp.push(Math.abs(left_ski[j] - right_ski[j]));
35             ++j;
36             k=j;
37         }
38         curvFieldAvg.push(getAverage(temp));
39     }
40     ++k;
41 }

```

4. Tehnologije sustava

4.1. ESP32-WROOM-32

ESP32-WROOM-32 je generički Wi-Fi + Bluetooth + Bluetooth LE MCU modul koji ima širok spektar aplikacija, od niskoenergetskih senzorskih mreža do najzahtjevnijih zadataka, kao što su enkodiranje glasa, prijenos (engl. *streaming*) glazbe i dekodiranje MP3 formata. U središtu ovog modula nalazi se ESP32-D0WDQ6 čip. Ugrađeni čip je dizajniran tako da bude skalabilan i prilagodljiv. Postoje dvije CPU jezgre koje se mogu pojedinačno kontrolirati, a frekvencija CPU-a je podesiva od 80 MHz do 240 MHz. Čip također ima koprocesor za nisku potrošnju energije koji se može koristiti umjesto glavnog CPU-a kako bi se uštedjela energija prilikom obavljanja zadataka koji ne zahtijevaju mnogo računalne snage, kao što je nadzor perifernih uređaja. ESP32 integrira bogat skup perifernih uređaja, od kapacitivnih senzora na dodir, SD kartičkog sučelja, Ethernet-a, visokobrzinskog SPI-a, UART-a, I2S-a i I2C-a.

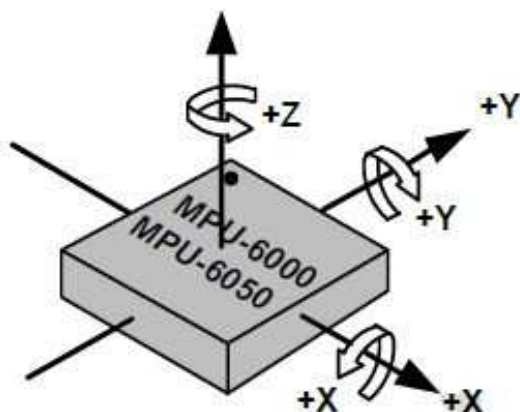
Integracija Bluetooth-a, Bluetooth LE-a i Wi-Fi-ja osigurava širok raspon aplikacija te omogućava da je modul svestran: korištenje Wi-Fi-ja omogućuje veliki doseg i izravnu vezu s Internetom, dok korištenje Bluetooth-a omogućuje korisniku praktično povezivanje s telefonom ili emitiranje niskoenergetskih signala za njegovo otkrivanje. Struja u stanju mirovanja ESP32 čipa je manja od 5 μ A, što ga čini prikladnim za aplikacije s napajanjem baterijom. Modul podržava brzinu prijenosa podataka do 150 Mbps i izlaznu snagu od 20 dBm na anteni. Operativni sustav odabran za ESP32 je freeRTOS. Podržano je i sigurno (kriptirano) ažuriranje (OTA), tako da korisnici mogu nadograditi svoje proizvode čak i nakon njihovog puštanja u produkciju uz minimalne troškove i trud.[5]

Tablica 4.1. ESP32-WROOM-32 Specifikacije [5]

Kategorije	Stavke	Specifikacije
Certifikacija	RF certifikacija	Certifikacija za ESP32-WROOM-32
	<i>Wi-Fi</i> certifikacija	<i>Wi-Fi</i> Alliance
	<i>Bluetooth</i> certifikacija	BQB
	Zelena certifikacija	RoHS/REACH
Test	Pouzdanost	HTOL/HTSL/UHAST/TCT/ESD
Wi-Fi	Protokoli	802.11 b/g/n (802.11n do 150 Mbps) A-MPDU i A-MSDU agregacija i 0.4 μ s interval zaštite
	Središnja frekvencijska raspona operativnog kanala	2412 ~ 2484 MHz
Bluetooth	Protokoli	<i>Bluetooth</i> v4.2 BR/EDR i <i>Bluetooth LE</i> specifikacija
	Radio	Odašiljač klase-1, klase-2 i klase-3 AFH NZIF prijemnik s osjetljivošću od -97 dBm
	Audio	CVSD i SBC
Sklopovlje	Modulska sučelja	SD kartica, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR, brojač impulsa, GPIO, kapacitivni dodirni senzor, ADC, DAC, dvosmjerno automobilsko sučelje (TWAI [®]), kompatibilno s ISO11898-1 (CAN specifikacija 2.0)
	Integrirani kristal	40 MHz kristal
	Integrirani SPI flash	4 MB
	Radni napon/Napajanje	3.0 V ~ 3.6 V
	Radna struja	Prosjek: 80 mA
	Minimalna struja koju isporučuje napajanje	500 mA
	Preporučeni radni temperaturni raspon	-40 °C ~ +85 °C
	Veličina paketa	18 mm × 25.5 mm × 3.10 mm
	Razina osjetljivosti na vlagu (MSL)	Razina 3

4.2. MPU6050

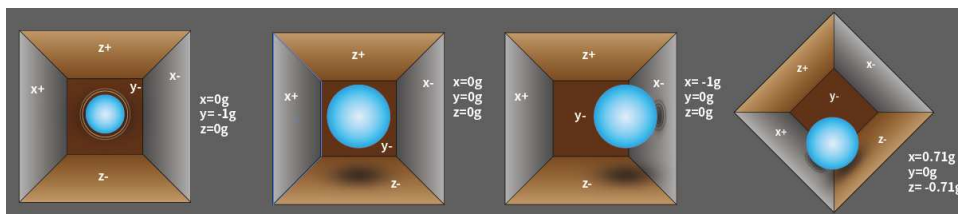
MPU-6050 je 6-osni (kombinira 3-osni žiroskop i 3-osni akcelerometar) uređaj za praćenje kretanja. Mogu se otkriti promjene u kretanju, ubrzanju i rotaciji. Često se koristi u robotici, kontrolerima za igranje i drugim elektroničkim uređajima koji zahtijevaju detekciju kretanja. Njegova visoka preciznost i niska cijena čine ga vrlo popularnim.[6]



Slika 4.1. Osi senzora MPU6050

4.2.1. Akcelerometar

Pravokutna kutija s malom kuglicom unutar nje može se zamisliti. Zidovi kutije izrađeni su od piezoelektričnih kristala. Kad god se kutija nagne, kuglica se prisilno kreće u smjeru nagiba zbog gravitacije. Zid s kojim kuglica dođe u dodir generira male piezoelektrične struje. Ukupno postoje tri para suprotnih zidova u pravokutniku, a svaki par odgovara jednoj osi u 3D prostoru: osi X, Y i Z. Na temelju struje proizvedene od piezoelektričnih zidova, moguće je odrediti smjer nagiba i njegovu veličinu.[6]



Slika 4.2. Primjer funkcioniranja akcelerometra

MPU6050 se može koristiti za detekciju ubrzanja na svakoj koordinatnoj osi (u stacionarnom stanju, ubrzanje na Z-osi je 1, dok su X i Y osi na 0). Ako je uređaj nagnut ili u

bestežinskom stanju, odgovarajuće očitavanje će se promijeniti.

Postoje četiri vrste mjernih raspona koje se mogu programski odabrati: $\pm 2g$, $\pm 4g$, $\pm 8g$ i $\pm 16g$ ($2g$ prema zadanim postavkama) koji odgovaraju svakoj preciznosti. Vrijednosti se kreću od -32768 do 32767 .

Očitavanje akcelerometra se pretvara u vrijednost ubrzanja mapiranjem očitavanja iz raspona očitavanja na mjerni raspon.

Ubrzanje = (Sirovi podaci osi akcelerometra / 65536 * puni mjerni raspon ubrzanja) g

Uzmimo X-os kao primjer, kada su sirovi podaci osi X akcelerometra 16384 i raspon je odabran kao $\pm 2g$:

Ubrzanje duž X osi = $(16384 / 65536 * 4) g = 1g$

4.2.2. Žiroskop

Žiroskopi funkcioniraju na principu Coriolisovog ubrzanja. Struktura nalik vilici koja se neprestano kreće naprijed-nazad može se zamisliti. Ova struktura je učvršćena piezoelektričnim kristalima. Kada se ovaj raspored nagne, kristali osjećaju silu u smjeru nagiba zbog inercije pokretne vilice. Kristali tada proizvode struju putem piezoelektričnog efekta, a ta se struja pojačava. Žiroskop također ima četiri vrste mjernih raspona: ± 250 , ± 500 , ± 1000 , ± 2000 . Metoda izračuna i ubrzanja je u osnovi dosljedna.

Formula za pretvaranje očitavanja u kutnu brzinu je sljedeća:

Kutna brzina = (Sirovi podaci osi žiroskopa / 65536 * puni mjerni raspon žiroskopa) $^{\circ}/s$

Na primjer, za X os, kada su sirovi podaci osi X akcelerometra 16384 , a raspon je $\pm 250^{\circ}/s$:

Kutna brzina duž X osi = $(16384 / 65536 * 500) ^{\circ}/s = 125 ^{\circ}/s$

4.3. Flutter

Flutter je otvoreni softverski razvojni paket (SDK) koji je kreirala tvrtka Google. Dizajniran je za izradu nativno kompajliranih aplikacija za mobilne uređaje (iOS, Android), web i

stolna računala iz jedinstvene baze koda. Lansiran u svibnju 2017., Flutter koristi programski jezik Dart, koji omogućuje brze razvojne cikluse te izražajno i fleksibilno korisničko sučelje.

Jedna od najistaknutijih značajki Flutter-a je njegova arhitektura temeljena na komponentama (engl. *widgets*). Sve u Flutter-u je komponenta (engl. *widget*), od osnovnih strukturnih elemenata poput gumba i teksta do složenih rasporeda. Ovaj dizajn omogućuje opsežnu prilagodbu i ponovnu upotrebu komponenti, promovirajući dosljedan i kohezivan dizajnerski jezik na različitim platformama.

Funkcija vruća zamjena (engl. *hot reload*) u Flutteru značajno povećava produktivnost programera omogućujući ažuriranja aplikacije u stvarnom vremenu bez potrebe za potpunim ponovnim pokretanjem. Ova sposobnost olakšava brzu iteraciju i eksperimentiranje, pojednostavljajući proces razvoja.

Još jedna ključna prednost Flutter-a je njegova izvedba. Kompiliranjem na izvorni ARM kod za iOS i Android, Flutter osigurava visoke performanse i glatke animacije, usporedive s nativnim aplikacijama. Osim toga, Flutter uključuje bogat skup unaprijed dizajniranih komponenta (engl. *widgets*) i tema, pridržavajući se i Material Design i Cupertino (iOS) standarda, osiguravajući nativni izgled na različitim platformama.

Ekosustav Flutter-a podržan je robusnom bibliotekom paketa, omogućujući programe- rima lako proširenje funkcionalnosti. Rastuća popularnost Flutter-a ogleda se u njegovoj aktivnoj zajednici i čestim ažuriranjima, čineći ga privlačnim izborom za moderan, multi- platformski razvoj aplikacija.

4.4. Spring

Spring olakšava razvoj Java aplikacija poduzeća (engl. *enterprise*). Pruža sve što je potrebno za korištenje Java jezika u okruženju poduzeća, uz podršku za Groovy i Kotlin kao alternativne jezike na JVM-u (Java virtualni stroj), te s fleksibilnošću za stvaranje raznih vrsta arhitektura ovisno o potrebama aplikacije.

Spring je otvorenog koda (engl. *open source*). Ima veliku i aktivnu zajednicu koja pruža kontinuirane povratne informacije temeljene na raznovrsnim stvarnim slučajevima upo-

trebe. To je pomoglo Spring-u da se uspješno razvija kroz vrlo dugo vrijeme.

Pojam Spring ima različita značenja u različitim kontekstima. Može se koristiti za referenciranje na sam projekt Spring okvir (engl. *framework*), što je njegovo izvorno značenje. Tijekom vremena, drugi Spring projekti su izgrađeni na vrhu Spring okvira. Najčešće, kada ljudi kažu Spring, misle na cijelu obitelj projekata.

Spring okvir je podijeljen na module. Aplikacije mogu odabrati koje module trebaju. U srcu su moduli osnovnog spremnika, uključujući model konfiguracije (engl. *configuration model*) i mehanizam za upravljanje ovisnostima (engl. *dependency injection*). Osim toga, Spring okvir pruža temeljnu podršku za različite arhitekture aplikacije, uključujući slanje poruka, transakcijske podatke i postojanost, te web. Također uključuje Servlet-temeljen Spring MVC web okvir i, paralelno, Spring WebFlux reaktivni web okvir. [7]

Spring Boot je modul Spring okvir koji omogućuje lagano kreiranje REST web servisa.

4.5. Kafka

4.5.1. Prijenos događaja

Tehnički gledano, prijenos događaja je praksa hvatanja podataka u stvarnom vremenu Iz izvora događaja kao što su baze podataka, senzori, mobilni uređaji, usluge u oblaku i softverske aplikacije u obliku tokova događaja; trajno pohranjivanje tih tokova događaja za kasnije dohvaćanje; manipuliranje, obrada i reagiranje na tokove događaja u stvarnom vremenu kao i retrospektivno; te usmjeravanje tokova događaja na različite ciljne tehnologije prema potrebi. Prijenos događaja tako osigurava kontinuirani tok i interpretaciju podataka kako bi prave informacije bile na pravom mjestu u pravo vrijeme.[8]

Prijenos događaja primjenjuje se na širok raspon slučajeva upotrebe u mnoštvu industrija i organizacija. Njegovi mnogi primjeri uključuju:

- Obrada plaćanja i financijskih transakcija u stvarnom vremenu, poput burzi, banaka i osiguranja.
- Praćenje i nadzor automobila, kamiona, flota i pošiljki u stvarnom vremenu, kao u logistici i automobilskoj industriji.

- Kontinuirano hvatanje i analiza podataka senzora s IoT uređaja ili druge opreme, kao u tvornicama i vjetroelektranama.
- Prikupljanje i trenutačno reagiranje na interakcije i narudžbe kupaca, kao u malo-prodaji, hotelskoj i putničkoj industriji te mobilnim aplikacijama.
- Praćenje pacijenata u bolničkoj skrbi i predviđanje promjena stanja kako bi se osigurala pravovremena liječenja u hitnim slučajevima.
- Povezivanje, pohranjivanje i dostupnost podataka proizvedenih od strane različitih odjela unutar tvrtke.
- Služenje kao temelj za podatkovne platforme, arhitekture vođene događajima i mikroservise.

4.5.2. Način rada

Kafka kombinira tri ključne mogućnosti, omogućujući implementaciju slučajeva upotrebe prijenosa događaja od početka do kraja:

- Objavljivanje (pisanje) i pretplata na (čitanje) tokove događaja, uključujući kontinuirani uvoz/izvoz podataka iz drugih sustava.
- Trajno i pouzdano pohranjivanje tokova događaja.
- Obrada tokova događaja dok se događaju ili retrospektivno.

Sve ove funkcionalnosti pružene su na distribuiran, visoko skalabilan, elastičan, otporan na greške i siguran način. Kafka se može implementirati na fizičkom hardveru, virtualnim strojevima i kontejnerima, te lokalno kao i u oblaku. Postoji mogućnost izbora između samostalnog upravljanja Kafka okruženjima i korištenja potpuno upravljanih usluga koje nude različiti dobavljači.[8]

4.5.3. Glavni koncepti i terminologija

Događaj bilježi činjenicu da se "nešto dogodilo" u svijetu ili poslovanju. U dokumentaciji se također naziva zapis ili poruka. Kada se čitaju ili pišu podatci u Kafki, to se radi u obliku

događaja. Konceptualno, događaj ima ključ, vrijednost, vremensku oznaku i opcionalno zaglavlje metapodataka. Primjer događaja:

- Ključ događaja: "Alice"
- Vrijednost događaja: "Izvršila uplatu od 200 dolara Bobu"
- Vremenska oznaka događaja: "25. lipnja 2020. u 14:06"

Proizvođači su one klijentske aplikacije koje objavljuju (pišu) događaje u Kafku, dok su potrošači one koje se pretplaćuju na (čitaju i obrađuju) te događaje. U Kafki su proizvođači i potrošači potpuno odvojeni i neovisni jedni o drugima, što je ključni dizajnerski element za postizanje visoke skalabilnosti po kojoj je Kafka poznata. Na primjer, proizvođači nikada ne trebaju čekati potrošače. Kafka pruža različite garancije kao što je mogućnost obrade događaja točno jednom.

Događaji su organizirani i trajno pohranjeni u temama. Vrlo pojednostavljeno, tema je slična mapi u datotečnom sustavu, a događaji su datoteke u toj mapi. Primjer imena teme mogao bi biti "uplate". Teme u Kafki su uvijek višestruki proizvođači i višestruki pretplatnici: tema može imati nula, jednog ili više proizvođača koji pišu događaje u nju, kao i nula, jednog ili više potrošača koji se pretplaćuju na te događaje. Događaji u temi mogu se čitati koliko god je potrebno, za razliku od tradicionalnih sustava za razmjenu poruka, događaji se ne brišu nakon konzumacije. Umjesto toga, definiramo koliko dugo Kafka treba zadržati događaje putem postavke konfiguracije za svaku temu, nakon čega će stari događaji biti odbačeni. Performanse Kafke su praktično konstantne u odnosu na veličinu podataka, tako da je dugoročno pohranjivanje podataka sasvim u redu.

Teme su particionirane, što znači da je tema raspoređena preko nekoliko "kanti" smještenih na različitim Kafka poslužiteljima. Ovo distribuirano postavljanje podataka vrlo je važno za skalabilnost jer omogućuje klijentskim aplikacijama da istovremeno čitaju i pišu podatke na/iz mnogih poslužitelja. Kada se novi događaj objavi u temi, zapravo se dodaje jednoj od particija te teme. Događaji s istim ključem događaja (npr. ID kupca ili vozila) zapisuju se u istu particiju, a Kafka jamči da će svaki potrošač određene tema-particije uvijek čitati događaje te particije točno istim redoslijedom kojim su napisani.

Kako bi podaci bili otporni na pogreške i visoko dostupni, svaka tema može biti replici-

rana, čak i između geo-regija ili podatkovnih centara, tako da uvijek postoji više poslužitelja (brokera) koji imaju kopiju podataka u slučaju da nešto pođe po zlu. Uobičajena produkcijska postavka je faktor replikacije od 3, tj. uvijek će postojati tri kopije podataka. Ova replikacija se izvodi na razini tema-particija.[8]

4.6. React

React je popularna otvorena JavaScript biblioteka koja se koristi za izradu korisničkih sučelja. Razvijen i održavan od strane Facebook-a, zajedno sa zajednicom pojedinačnih programera i tvrtki, React je značajno utjecao na način na koji se moderne web aplikacije izgrađuju.

React aplikacije se grade pomoću komponenti, koje su građevni blokovi korisničkog sučelja. Komponenta u Reactu je samostalni modul koji prikazuje neki izlaz. Komponente mogu biti bazirane na klasama ili funkcijama. Funkcijske komponente su jednostavnije, a s uvođenjem hook-ova mogu upravljati stanjem, što ih čini moćnijima i često poželjnijima.

React koristi JSX, proširenje sintakse za JavaScript, koje omogućava programerima da pišu kod sličan HTML-u unutar JavaScript-a. Ovaj sintaktički šećer poboljšava čitljivost koda i pomaže u opisu strukture korisničkog sučelja na deklarativniji način.

Jedna od ključnih značajki React-a je Virtualni DOM. Umjesto direktne manipulacije DOM-om, React stvara laganu reprezentaciju DOM-a u memoriji. Kada se stanje objekta promijeni, React prvo ažurira Virtualni DOM, a zatim učinkovito ažurira stvarni DOM kako bi odgovarao Virtualnom DOM-u. Ovaj proces minimizira troškove performansi povezane s direktnom manipulacijom DOM-om.

State i props su osnovni koncepti u React-u. State predstavlja promjenjive podatke koje komponenta može održavati i upravljati. Props skraćeno od svojstva (engl. *properties*) su za čitanje atributa koji se prenose iz roditeljskih komponenti u dječje komponente. Props omogućuju ponovnu upotrebu komponentata prijenosom dinamičkih podataka i ponašanja.

React komponente prolaze kroz životni ciklus montaže, ažuriranja i demontaže. Klase komponenti nude metode životnog ciklusa poput *componentDidMount*, *componentDidUp-*

date i *componentWillUnmount* koje omogućavaju programerima da se uključe u određene trenutke životnog ciklusa komponente kako bi izvršili kod u specifičnim vremenima. Funkcionalne komponente koriste hook-ove poput *useEffect* za postizanje sličnih rezultata.

React je revolucionirao razvoj klijentskih aplikacija svojom arhitekturom baziranom na komponentama i učinkovitim prikazivanjem korištenjem Virtualnog DOM-a. Njegova snažna podrška zajednice, bogat ekosustav alata i biblioteka čine ga svestranim izborom za izradu modernih web aplikacija. Usvajanjem koncepata poput hook-ova, konteksta i konkurentnog načina rada (engl. *concurrent mode*), React nastavlja evoluirati, nudeći programerima moćne alate za kreiranje skalabilnih i visokoučinkovitih korisničkih sučelja.

4.7. Bluetooth

Bluetooth je bežična komunikacijska tehnologija kratkog dometa koja se koristi za razmjenu podataka između uređaja. Uz Wi-Fi, najkorištenija je metoda prijenosa podataka na tržištu te se uglavnom koristi u prijenosnim uređajima, računalima, pametnim satovima, kontrolerima, audio uređajima i ugradbenim uređajima.

4.7.1. Bluetooth Classic

Bluetooth Classic radio, također nazivan Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR), je radio s niskom potrošnjom energije koji prenosi podatke preko 79 kanala u 2,4 GHz neli-cenciranom industrijskom, znanstvenom i medicinskom (ISM) frekvencijskom pojasu. Podržavajući komunikaciju uređaja od točke do točke, Bluetooth Classic se uglavnom koristi za omogućavanje bežičnog prijenosa zvuka i postao je standardni radio protokol iza bežič-nih zvučnika, slušalica i zabavnih sustava u automobilima. Bluetooth Classic radio također omogućuje aplikacije za prijenos podataka, uključujući mobilno ispisivanje/printanje.[9]

4.7.2. Bluetooth Low Energy (LE)

Bluetooth Low Energy (LE) radio je dizajniran za rad s vrlo niskom potrošnjom energije. Prenoseći podatke preko 40 kanala u 2,4 GHz neli-cenciranom ISM frekvencijskom pojasu, Bluetooth LE radio pruža programerima veliku fleksibilnost za izradu proizvoda koji zadovoljavaju jedinstvene zahtjeve povezivanja na njihovom tržištu. Bluetooth LE podržava više komunikacijskih topologija, proširujući se od točke do točke do emitiranja i, najnovije,

mrežnog umrežavanja mreža (engl. *mesh*), omogućujući Bluetooth tehnologiji podršku za stvaranje pouzdanih, velikih mreža uređaja. Iako je u početku bio poznat po svojim mogućnostima komunikacije između uređaja, Bluetooth LE se sada također široko koristi kao tehnologija za pozicioniranje uređaja. Bluetooth LE sada uključuje značajke koje omogućuju jednom uređaju da odredi prisutnost, udaljenost i smjer drugog uređaja.[9]

	Bluetooth Low Energy (LE)	Bluetooth Classic
Frekvencije	2.4GHz ISM Band (2.402 – 2.480 GHz korišteno)	2.4GHz ISM Band (2.402 – 2.480 GHz korišteno)
Kanali	40 kanala s razmakom od 2 MHz (3 reklamirajuća kanala / 37 podatkovnih kanala)	79 kanala s razmakom od 1 MHz
Korištenje kanala	<i>Frequency-Hopping Spread Spectrum (FHSS)</i>	<i>Frequency-Hopping Spread Spectrum (FHSS)</i>
Modulacija	GFSK	GFSK, $\pi/4$ DQPSK, 8DPSK
Brzina prijenosa podataka	LE 2M PHY: 2 Mb/s LE 1M PHY: 1 Mb/s LE Coded PHY (S=2): 500 Kb/s LE Coded PHY (S=8): 125 Kb/s	EDR PHY (8DPSK): 3 Mb/s EDR PHY ($\pi/4$ DQPSK): 2 Mb/s BR PHY (GFSK): 1 Mb/s
Tx snaga	≤ 100 mW (+20 dBm)	≤ 100 mW (+20 dBm)
Rx osjetljivost	LE 2M PHY: ≤ -70 dBm LE 1M PHY: ≤ -70 dBm LE Coded PHY (S=2): ≤ -75 dBm LE Coded PHY (S=8): ≤ -82 dBm	≤ -70 dBm
Transport podataka	Asinkrono, sinkrono, izosinkrono	Asinkrono, sinkrono
Komunikacijske topologije	<i>Point-to-Point, Broadcast, Mesh</i>	<i>Point-to-Point</i>
Značajke pozicioniranja	<i>Prisutnost: Oglašavanje Smjer: Određivanje smjera (AoA/AoD) Udaljenost: RSSI, HADM (dolazno)</i>	-

Tablica 4.2. Usporedba Bluetooth Low Energy (LE) i Bluetooth Classic [9]

4.8. HTTP

HTTP (HyperText Transfer Protocol) je temelj komunikacije podataka na Internetu. To je protokol aplikacijskog sloja i koristi se za prijenos hipermedijskih dokumenata, kao što je HTML. HTTP je stateless protokol, što znači da se svaki zahtjev klijenta prema poslužitelju obrađuje neovisno, bez ikakvog znanja o prethodnim zahtjevima.

Ključni koncepti HTTP-a:

1. Model klijent-poslužitelj

HTTP protokol temelji se na modelu klijent-poslužitelj. Klijent (često web preglednik) šalje HTTP zahtjev poslužitelju, koji poslužuje resurse (poput web stranica, slika, videozapisa, itd.).

Poslužitelj obrađuje zahtjev i šalje natrag HTTP odgovor, koji uključuje statusne informacije o zahtjevu i može uključivati traženi sadržaj.

2. Zahtjevi i odgovori

Zahtjev: Klijent ga šalje i sastoji se od linije zahtjeva (metoda, putanja resursa i HTTP verzija), zaglavlja (metapodaci) i opcionalno tijela (podataka).

Odgovor: Poslužitelj ga šalje i uključuje statusnu liniju (HTTP verzija, statusni kod i fraza), zaglavlja i opcionalno tijelo (podatke).

3. Zahtjevi i odgovori

Metode:

GET: Traži određeni resurs.

POST: Podnosi podatke na obradu na određeni resurs.

PUT: Zamjenjuje podatke ciljanog resursa s tijelom zahtjeva.

DELETE: Uklanja određeni resurs.

HEAD: Isto kao GET, ali prenosi samo statusnu liniju i zaglavlja.

OPTIONS: Opisuje komunikacijske opcije za ciljani resurs.

PATCH: Primjenjuje djelomične izmjene na ciljani resurs.

4. Statusni kodovi

1xx: Informativni - Ovi kodovi označavaju da je zahtjev primljen i da se proces nastavlja.

2xx: Uspjeh - Ovi kodovi označavaju da je zahtjev uspješno primljen, razumjen i prihvaćen.

3xx: Preusmjerenje - Ovi kodovi označavaju da korisnički agent mora poduzeti dodatne radnje kako bi ispunio zahtjev.

4xx: Pogreška klijenta - Ovi kodovi označavaju da zahtjev sadrži lošu sintaksu ili ga nije moguće ispuniti.

5xx: Pogreška servera - Ovi kodovi označavaju da server nije uspio ispuniti valjani zahtjev.

5. Zaglavlja

HTTP zaglavlja pružaju ključne informacije o zahtjevu ili odgovoru, ili o objektu poslanom u tijelu poruke.

Uobičajena zaglavlja uključuju Content-Type, Authorization, User-Agent, Accept i Host.

6. Kolačići

Mali dijelovi podataka poslani s poslužitelja i pohranjeni na klijentovoj strani za upravljanje stanjem između HTTP zahtjeva, kao što su korisničke sesije.

7. Sigurnost

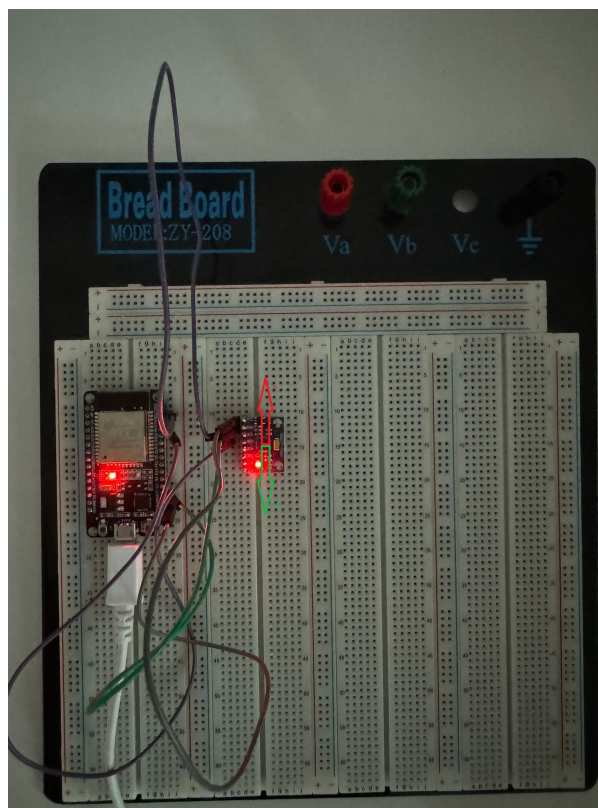
HTTP se može osigurati korištenjem HTTPS (HTTP Secure), koji šifrira podatke između klijenta i poslužitelja koristeći TLS (Transport Layer Security).

5. Testiranje sustava

Prije početka testiranja je potrebno upisati IP adresu poslužiteljskog servisa u aplikaciju 2.5. Prilikom testiranja svih slučajeva sustava trenutna informacija o nagibu skija se nalazi na mobilnom uređaju 2.6.

Ako se izgubi Bluetooth veza između jedne od skija potrebno je stisnuti na gumb **Re-connect**.

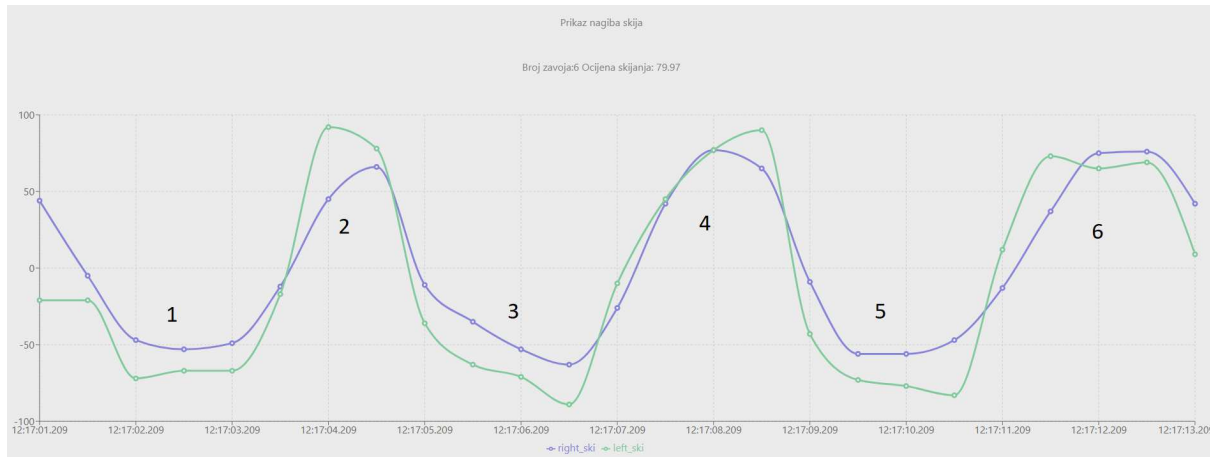
Prikaz smjera nagnjanja senzora je prikazan na 5.1., gdje zelena strjelica prikazuje pozitivan smjer (kut), a crvena strjelica negativan smjer (kut).



Slika 5.1. Smjer nagnjanja senzora

5.1. Simulacija realističnog skijanja

Kod simulacije realističnog gibanja skija, senzori su rukama pomjerani tako da simuliraju pravo skijanje.



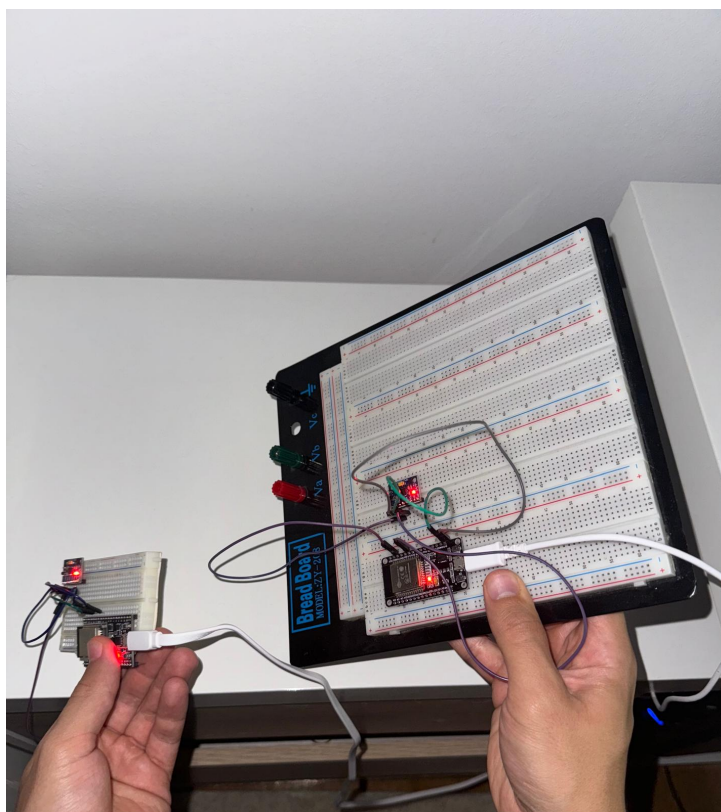
Slika 5.2. Realistično ručno kretanje skija

Kod realističnog ručnog kretanja obje skije 5.2. algoritam određuje da je broj zavoja 6 što je točno i daje prikladnu ocjenu 79.97. Svaki zavoj koji je algoritam uzeo u obzir na 5.2. je numeriran. Također može se zaključiti da su u prvom zavoju skije nagnute u lijevu stranu i što znači da skijaš skreće lijevo i započinje periodično vijuganje.

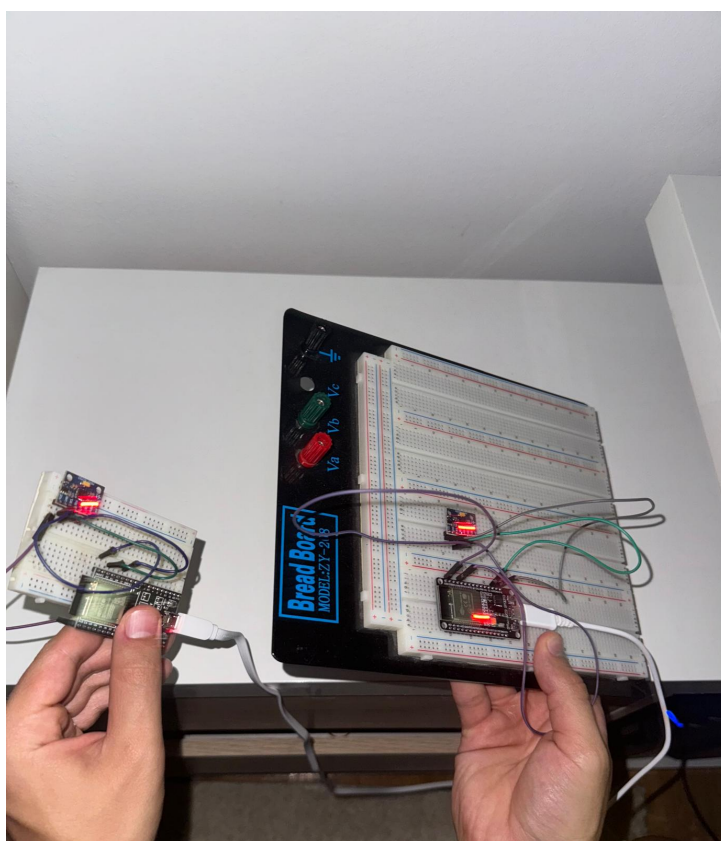
Kod ručnog kretanja 5.2. se može uočiti ljudska greška kašnjenja nagibanja skija i različitosti u kutevima nagiba.

Kako se radi o relativno periodičnom vijuganju može se iz grafa 5.2. izračunati okvirni prosječni period/vrijeme jednog zavoja:

$$T = \frac{2000ms + 1350ms + 2200ms + 1500ms + 2000ms + 2100ms}{6} = 1858.33ms$$

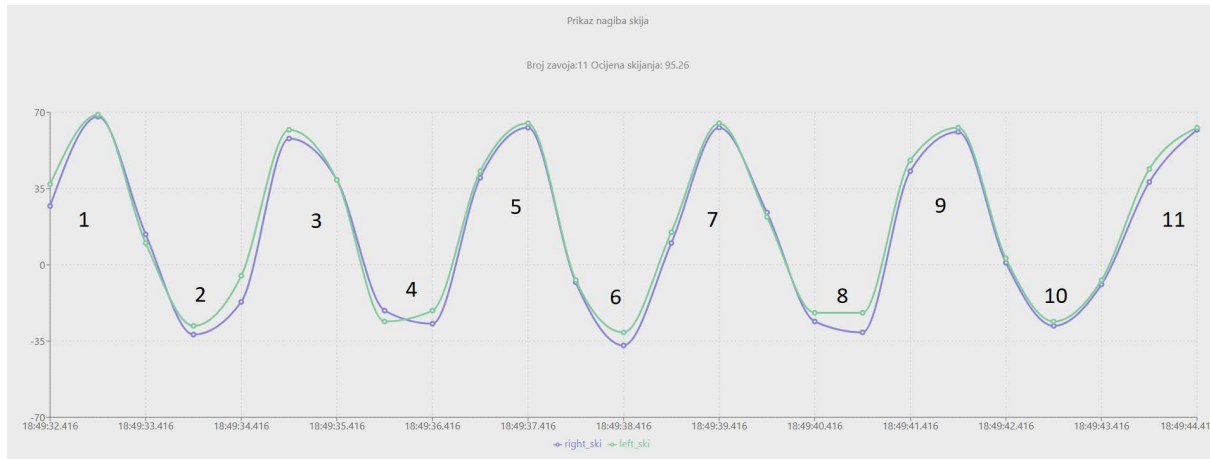


Slika 5.3. Lijevi zavoј ručnog kretanja skija



Slika 5.4. Desni zavoј ručnog kretanja skija

5.2. Simulacija idealnog skijanja



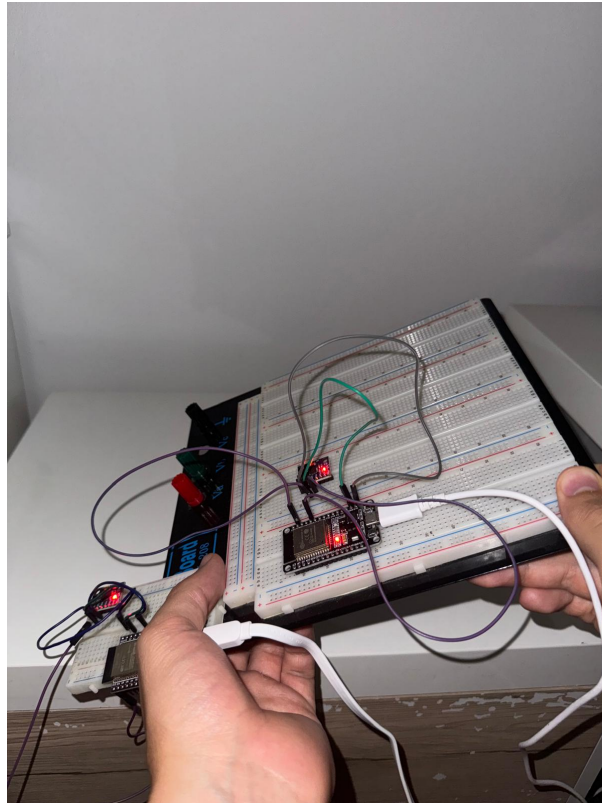
Slika 5.5. Idealno skijanje na istoj ravlini

Kod idealnog kretanja obje skije na istoj ravlini 5.5. algoritam određuje da je broj zavoja 11 što je točno i daje dobru ocjenu 95.26. Svaki zavoj koji je algoritam uzeo u obzir na 5.5. je numeriran. U prvom zavoju skije nagnute u desnu stranu i što znači da skijaš skreće desno i započinje periodično vijuganje.

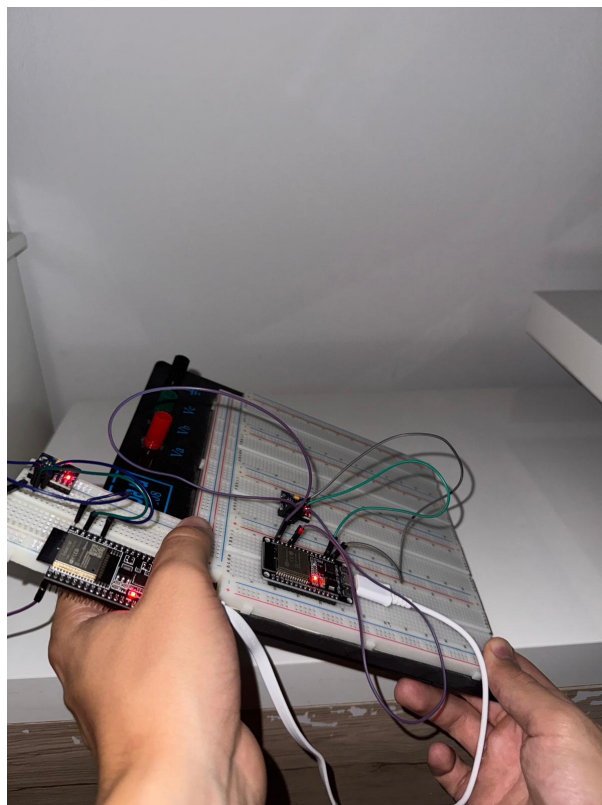
Rub površine jednog sklopa je poslužio kako bi senzori bili na istoj površini kao što se može uočiti na slikama 5.6. i 5.7.

Radi se o relativno periodičnom vijuganju i može se iz grafa 5.5. izračunati okvirni prosječni period/vrijeme jednog zavoja (pošto zavoji 1 i 11 nisu potpuni, izbačeni su iz izračuna):

$$T = \frac{1000ms + 1200ms + 900ms + 1250ms + 950ms + 1350ms + 950ms + 1300ms + 1100ms}{9} = 1111.11ms$$

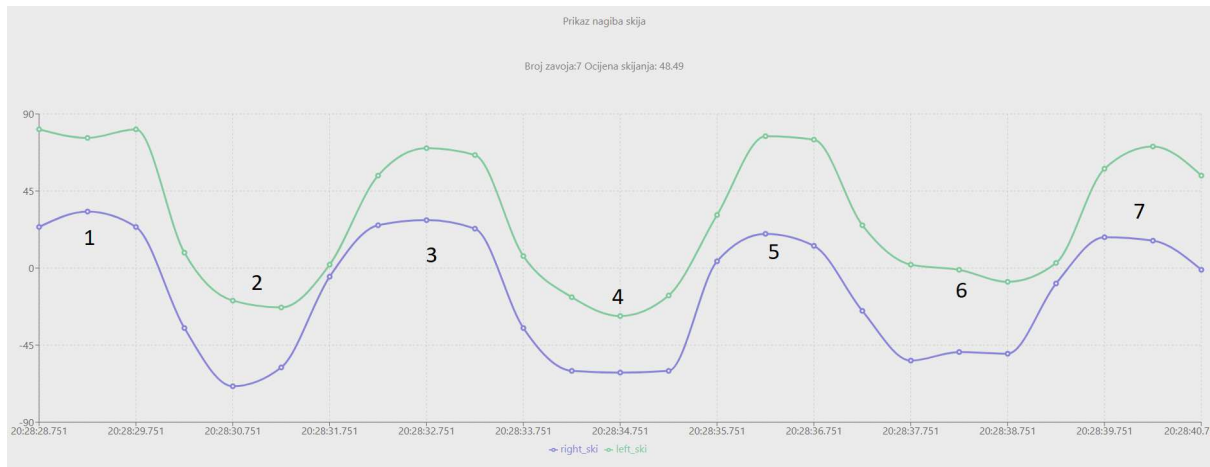


Slika 5.6. Lijevi zavoj ručnog kretanja skija na istoj površini



Slika 5.7. Desni zavoj ručnog kretanja skija na istoj površini

5.3. Simulacija sistematično slabijeg naginjanja unutar- nje skije

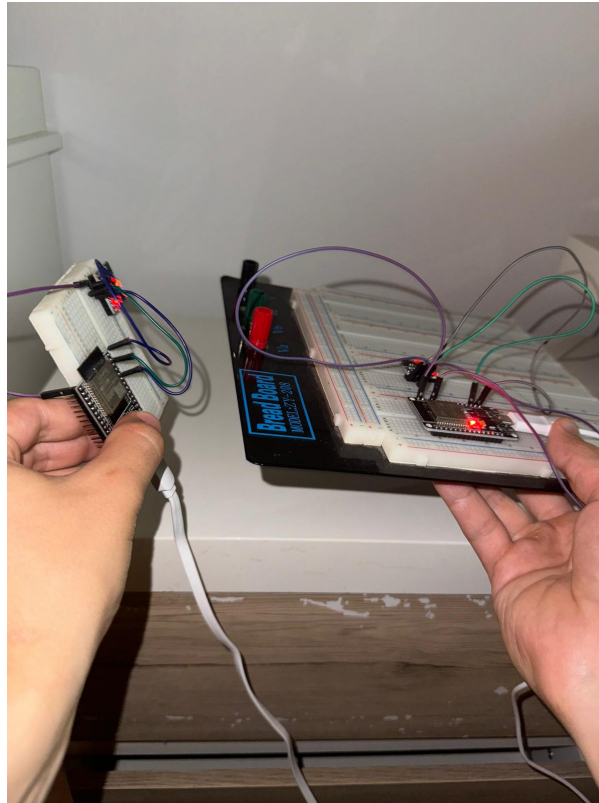


Slika 5.8. Simetrično slabije naginjanje unutarnje skije

Kod simetrično slabijeg naginjanja unutarnje skije 5.8. algoritam određuje da je broj zavoja 7 što je točno i daje prikladnu ocjenu 48.49. Svaki zavoj koji je algoritam uzeo u obzir na 5.8. je numeriran. Također može se zaključiti da su u prvom zavoju skije nagnute u desnu stranu i što znači da skijaš skreće desno i započinje periodično vijuganje.

Radi se o relativno periodičnom vijuganju i može se iz grafa 5.8. izračunati okvirni prosječni period/vrijeme jednog zavoja (pošto zavoji 1 i 7 nisu potpuni, izbačeni su iz izračuna):

$$T = \frac{1400ms + 2000ms + 1700ms + 2300ms + 1500ms}{5} = 1780ms$$



Slika 5.9. Desni zavoj sa slabijim nagnjanjem unutarnje skije

5.4. Simulacija kašnjenja rubljenja unutarnje skije u početku zavoja



Slika 5.10. Kašnjenje rubljenja unutarnje skije

Kod kašnjenja rubljenja unutarnje skije 5.10. algoritam određuje da je broj zavoja 5 što je točno i daje prikladnu ocjenu 68.99. Svaki zavoj koji je algoritam uzeo u obzir na 5.10. je

numeriran. Također može se zaključiti da su u prvom zavoju skije nagnute u desnu stranu što znači da skijaš skreće desno i započinje periodično vijuganje.

Može se zaključiti da postoji kašnjenje nagibanja jedne skije od 500ms.

Radi se o relativno periodičnom vijuganju i može se iz grafa 5.10. izračunati okvirni prosječni period/vrijeme jednog zavoja (pošto zavoji 1 i 5 nisu potpuni, izbačeni su iz izračuna):

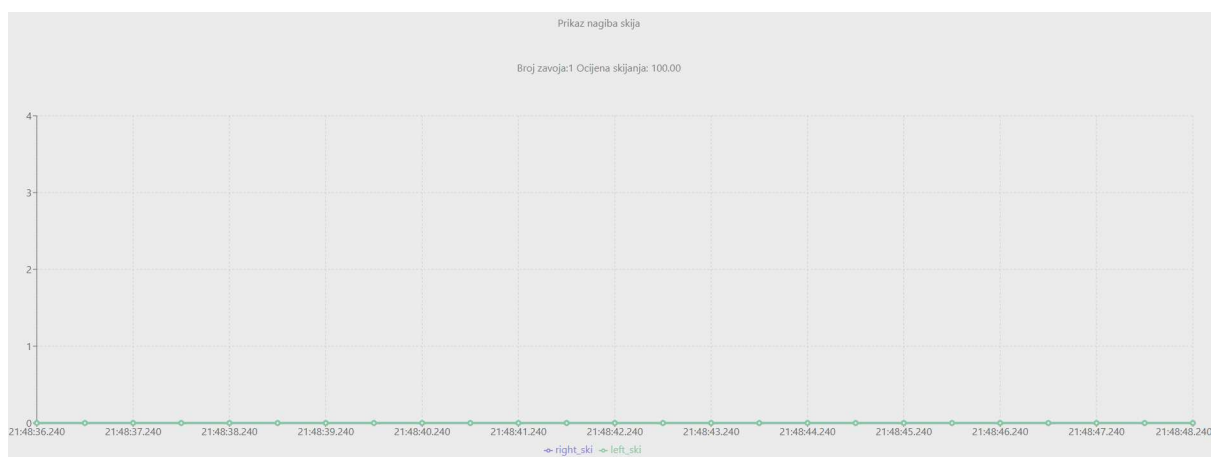
$$T = \frac{3500ms + 3700ms + 2800ms}{3} = 3333.33ms$$

5.5. Testiranje pozicija ekstrema

Pozicije ekstrema skija su kada su obje skije paralelno i kada su okomite.

U paralelnoj poziciji je najveća skijaška vještina, u ovom konkretnom slučaju algoritam daje vrijednost 100 i broj zavoja je 1.

Zanimljiva obzervacija je da algoritam očitava da je broj zavoja 1 dok se zavoji ne izvode, a to je zato što je u algoritmu određeno da ako je kut između skija 0 onda se izvodi savršen zavoj.



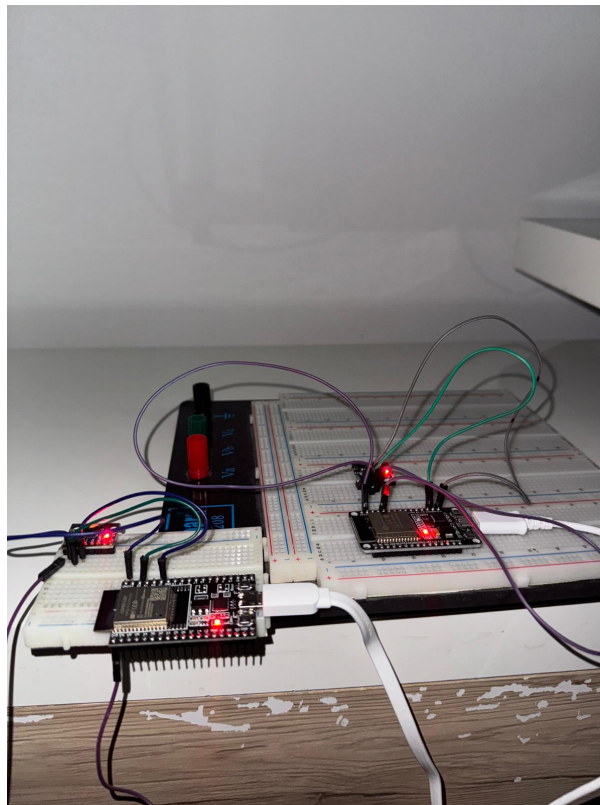
Slika 5.11. Graf kada su obje skije na istoj ravnini

U sljedećem slučaju 5.12. kada su skije okomite algoritam dobro zaključuje i daje vrijednost 0.00. Broj zavoja je 1 jer algoritam pretpostavlja da je skijaš cijelo vrijeme u kontinuiranom jako lošem zavoju.

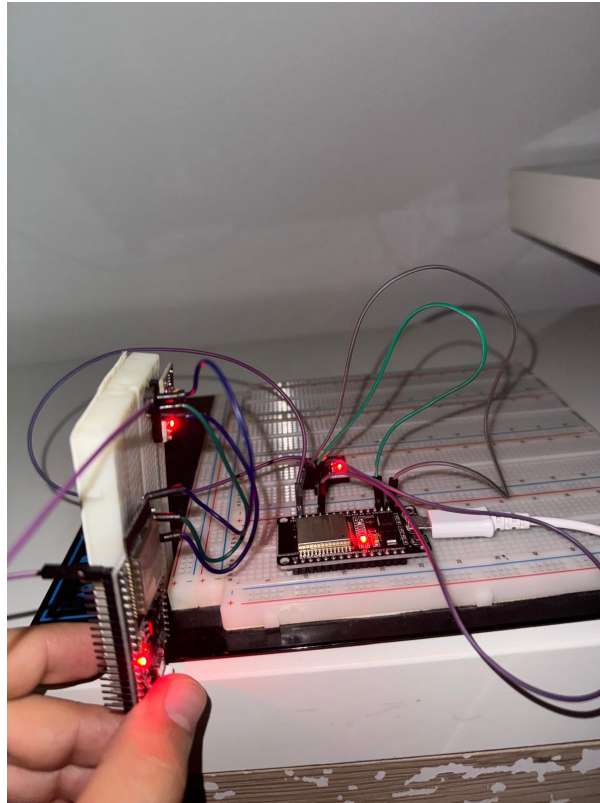


Slika 5.12. Graf okomitih skija

U oba slučaja ne postoji nikakvo vijuganje tako da nema smisla mjeriti period.



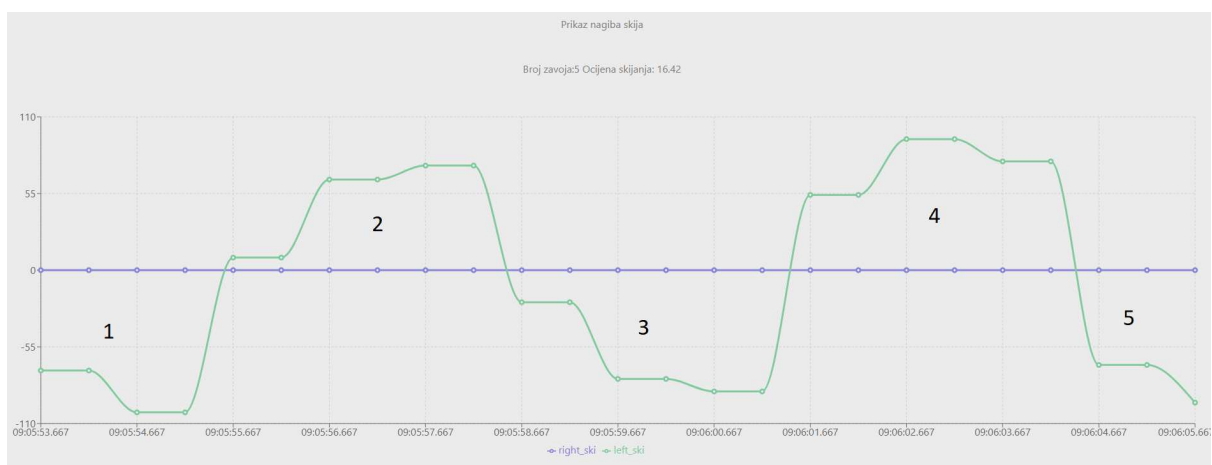
Slika 5.13. Skije na istoj ravnini



Slika 5.14. Okomite skije

5.6. Simulacija jedne skije

U narednim slikama je prikazano kretanje jedne skije dok je druga stacionarna, ova situacija se može usporediti sa situacijom kada skijaš skija na samo jednoj skiji.



Slika 5.15. Grafa kretanja jedne skije

Kod simulacije jedne skije 5.15. algoritam određuje da je broj zavoja 5 što je točno i daje

prikladnu ocjenu 16.42. Svaki zavoј koji je algoritam uzeo u obzir na 5.15. je numeriran. Također se može zaključiti da u prvom zavoju skija koja se kreće je nagnuta u lijevu stranu i skijaš skreće lijevo.

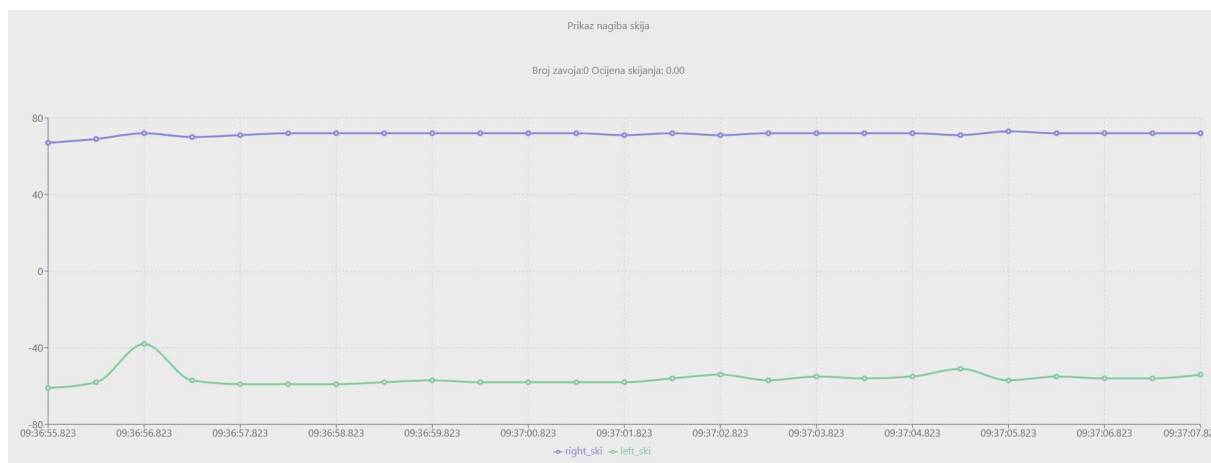
Radi se o relativno periodičnom vijuganju i može se iz grafa 5.15. izračunati okvirni prosječni period/vrijeme јednog zavoја (pošto su zavoји 1 i 5 odrezani, izbačeni su iz izračuna):

$$T = \frac{2900ms + 2900ms + 3000ms}{3} = 2933.33ms$$

5.7. Simulacija unakrsnog kretanja skija

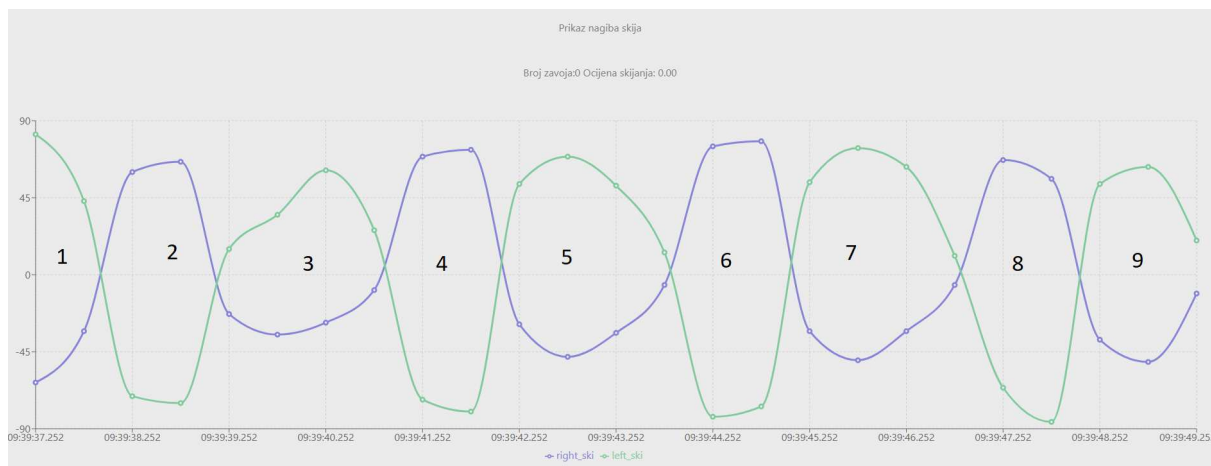
Prvi slučaj je kada su skije u plužnom položaju. Kod simulacije pluga 5.16. algoritam određuje da je broj zavoја 0 što je točno i daje prikladnu ocjenu 0.

Naime kod plužnog skijanja je također moguće raditi zavoје ali kao što se može ustanoviti trenutni algoritam to ne može zaključiti.

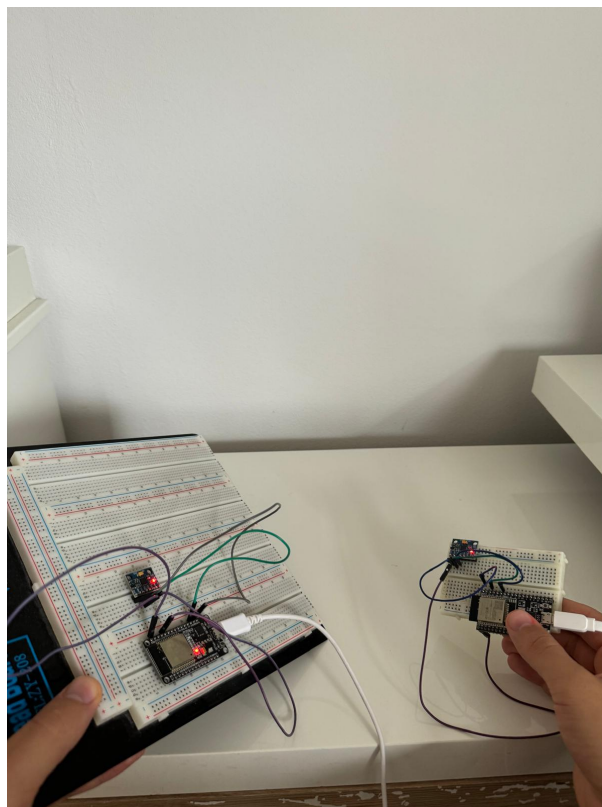


Slika 5.16. Graf plužnog kretanja skija

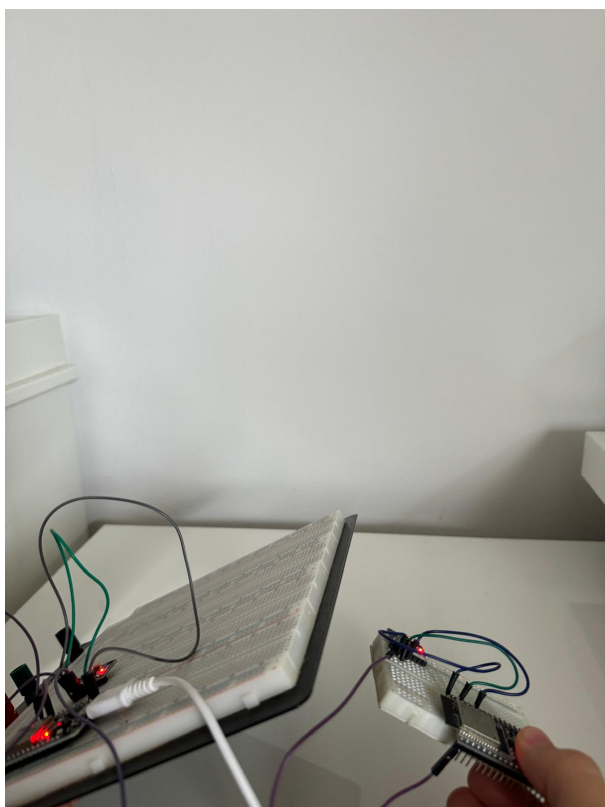
Prilikom unakrsnog skijanja 5.17. algoritam je prepoznao 0 zavoја i skijašku vještinu od 0. Unakrsno skijanje zapravo nije temeljna skijaška vještina tako da trenutni primjer je samo zanimljiv prikaz kako algoritam funkcionira.



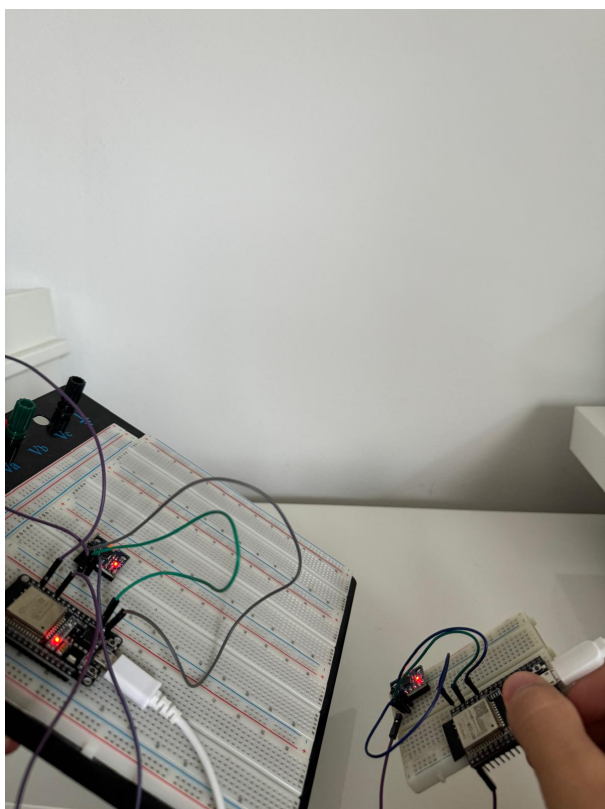
Slika 5.17. Graf unakrsnog kretanja skija



Slika 5.18. Plug



Slika 5.19. Unakrsne skije



Slika 5.20. Unakrsne skije

6. Daljnji rad i razmatranje

Temeljna funkcionalnost ovog projekta je ostvarena, a postoje mnogi načini za unapređenje. Kod sloja senzora mogu se dodati i drugi senzori, kao npr. uložak za pancericu koji bi mjerio pritisak stopala na određene točke u pancERICI i prema tim podacima omogućio dodatnu analizu. Ovi senzori mogu pružiti detaljne informacije o raspodjeli težine i pritisku, što skijašima omogućuje bolje razumijevanje kako njihove tehnike i pokreti utječu na performanse i udobnost.

Mobilna aplikacija može se unaprijediti za dodatnu analizu podataka ovisno o mjenjima sa senzora. Aplikacija može uključivati različita stanja aktivnosti u skijanju, npr. mirovanje, na gondoli, skijanje. Ova funkcionalnost omogućuje korisnicima da dobiju precizne podatke i analize specifične za različite faze skijaškog dana, pružajući dublje uvide u performanse i omogućujući optimizaciju treninga i odmora.

Daljnja nadogradnja sustava može uključivati logiranje korisnika i spremanje podataka o skijanju u bazu, što omogućuje detaljnije analize podataka i natjecanja u skijaškoj formi. Korištenjem ovakve baze podataka, skijaši mogu pratiti svoj napredak kroz vrijeme, uspoređivati performanse s prijateljima ili drugim korisnicima te sudjelovati u virtualnim natjecanjima, što dodatno motivira na kontinuirano poboljšanje.

Kućište senzora je također važna stavka sustava o kojoj treba razmisliti, uključujući način postavljanja tog kućišta na pancericu. Razvoj ergonomskog i izdržljivog kućišta koje neće ometati skijaša tijekom aktivnosti ključan je za dugotrajnu upotrebu i pouzdanost senzora. Potrebno je uzeti u obzir različite klimatske uvjete i fizičke zahtjeve skijanja kako bi se osiguralo da kućište može izdržati te uvjete bez kompromitiranja funkcionalnosti senzora.

Uz navedene nadogradnje, moguće je implementirati dodatne značajke kao što su in-

tegracija sa društvenim mrežama, omogućujući skijašima da dijele svoje rezultate i postignuća s prijateljima i pratiteljima, te razvoj personaliziranih trening programa temeljenih na prikupljenim podacima. Sve ove inovacije doprinose stvaranju naprednog, sveobuhvatnog sustava koji ne samo da prati i analizira parametre skijanja, već i aktivno doprinosi poboljšanju skijaških vještina i užitku u sportu skijanja.

7. Zaključak

Zaključuje se da ovaj projekt obuhvaća mnoge sfere programiranja i da nije jednostavno razviti kompleksan sustav za praćenje parametara skijaša. Svaki sloj arhitekture nosi svoje izazove te je potrebno imati dobre uvide u tehnologije koje se koriste. Uzimajući u obzir raznolikost senzora, algoritama za obradu podataka te integraciju softverskih i hardverskih komponenti, razvoj ovakvog sustava zahtijeva temeljito planiranje i preciznu implementaciju.

Jedan od ključnih izazova u ovom projektu je obrada velike količine podataka u stvarnom vremenu. Sustav mora biti sposoban prikupljati podatke sa senzora postavljenih na skijašu i analizirati te podatke kako bi se dobili korisni uvidi u performanse skijaša. Ovo uključuje razvoj algoritma za filtriranje i analizu podataka, kao i optimizaciju performansi sustava kako bi se osigurala brza i točna povratna informacija.

Kroz daljnji razvoj i nadogradnju ovog rada postoji mogućnost konkuriranja Carv sustavu koji je trenutno na tržištu. Carv sustav se već dokazao kao kvalitetno rješenje za praćenje parametara skijanja, ali i ovaj sustav ima potencijal ponuditi inovativne funkcionalnosti i poboljšanja koja bi mogla privući korisnike.

Razvoj ovakvog sustava zahtijeva multidisciplinarni pristup, uključujući stručnjake za razvoj softvera, podatkovnu znanost, biomehaniku i sportski trening. Uz kontinuirani rad i inovacije, može se stvoriti sustav koji će značajno unaprijediti iskustvo skijanja i doprinijeti poboljšanju performansi skijaša na svim razinama.

Literatura

- [1] Recharts Group, <https://recharts.org/en-US/>.
- [2] Espressif Systems, <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/i2c.html>.
- [3] —, <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-guides/bluetooth.html>.
- [4] Bluetooth SIG, <https://www.bluetooth.com/wp-content/uploads/Files/Specification/HTML/Core-54/out/en/host/generic-attribute-profile--gatt-.html>.
- [5] Espressif Systems, https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.
- [6] SunFounder, https://docs.sunfounder.com/projects/ultimate-sensor-kit/en/latest/components_basic/05-component_mpu6050.html.
- [7] Broadcom, <https://docs.spring.io/spring-framework/reference/overview.html>.
- [8] Apache Software Foundation, <https://kafka.apache.org/intro>.
- [9] Bluetooth SIG, <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>.

Sažetak

Sustav za praćenje kinematičkih parametara skijaša

Antonio Sabljic

Volja za poboljšanjem skijaške forme, inspirirana već postojećim sustavom Carv za skijanje, potaknula je ovaj rad. Ovaj rad opisuje izradu i djelovanje sustava za praćenje nagiba skija. Detaljno je opisan svaki sloj sustava: sloj osjetila, mobilna aplikacija, oblak i vizualizacija podataka. U sloju osjetila koriste se napredni senzori postavljeni na pancericice za precizno mjerenje kuta nagiba skija. Mobilna aplikacija omogućuje prikupljanje i inicijalnu obradu podataka, te korisniku pruža trenutne povratne informacije. Oblak služi za pohranu i daljnju analizu prikupljenih podataka, omogućujući pristup podacima s bilo koje lokacije. Vizualizacija podataka prikazuje rezultate u realnom vremenu putem intuitivnih grafičkih sučelja, omogućujući skijašima kontinuirano praćenje i poboljšanje njihove tehnike. Također su opisane tehnologije koje su se koristile u svakom sloju, uključujući bežične komunikacijske protokole i algoritme za obradu podataka.

Ključne riječi: skijanje; sloj osjetila; mobilna aplikacija; oblak; nagib skija

Abstract

A system for monitoring the kinematic parameters of skiers

Antonio Sabljic

The desire to improve skiing form, inspired by the existing Carv system for skiing, has motivated this work. This paper describes the development and operation of a system for tracking the angle of skis. Each layer of the system is described in detail: the sensory layer, the mobile application, the cloud, and data visualization. The sensory layer uses advanced sensors placed on ski boots for precise measurement of the ski angle. The mobile application allows for data collection and initial processing, providing the user with immediate feedback. The cloud is used for storing and further analyzing the collected data, enabling access from any location. Data visualization displays the results in real-time through intuitive graphical interfaces, allowing skiers to continuously monitor and improve their technique. The technologies used in each layer are also described, including wireless communication protocols and data processing algorithms.

Keywords: skiing; sensor layer; mobile application; cloud; ski tilt

Privitak A: The Code