

Sustav za prepoznavanje aktivnosti osobe u kućanstvu

Kovačić, Mislav

Master's thesis / Diplomski rad

2025

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:168:383571>

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

Download date / Datum preuzimanja: **2025-03-14**



Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 703

**SUSTAV ZA PREPOZNAVANJE AKTIVNOSTI OSOBE U
KUĆANSTVU**

Mislav Kovačić

Zagreb, veljača 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 703

**SUSTAV ZA PREPOZNAVANJE AKTIVNOSTI OSOBE U
KUĆANSTVU**

Mislav Kovačić

Zagreb, veljača 2025.

DIPLOMSKI ZADATAK br. 703

Pristupnik: **Mislav Kovačić (0036519935)**
Studij: Računarstvo
Profil: Programsko inženjerstvo i informacijski sustavi
Mentor: izv. prof. dr. sc. Igor Čavrak

Zadatak: **Sustav za prepoznavanje aktivnosti osobe u kućanstvu**

Opis zadatka:

Praćenje i prepoznavanje aktivnosti osobe u kućanstvu postaje sve važnije, posebno u kontekstu pametnih kućanstava i skrbi o starijim osobama. Kako populacija stari, postoji potreba za održavanjem kvalitete života i sigurnosti starijih osoba u njihovom domu. Ovakvi sustavi uključuju praćenje aktivnosti i prepoznavanje hitnih stanja koja zahtijevaju intervenciju. Osnovni problemi u ovom području uključuju izazove u očuvanju privatnosti korisnika i točnost prepoznavanja aktivnosti. Tema ovog diplomskog rada je razvoj raspodijeljenog sustava za praćenje i prepoznavanje aktivnosti osoba u kućanstvu temeljenog na mmWave osjetilima i mikrokontrolerima, uz naglasak na očuvanje privatnosti nadgledanih osoba. Potrebno je proučiti problematiku praćenja i prepoznavanja aktivnosti osoba u kućanstvu te specifične izazove vezane uz skrb o starijim osobama u njihovu domu. Identificirati i analizirati svojstva metoda prepoznavanja stanja osobe u kontekstu očuvanja privatnosti. Predložiti arhitekturu i građu raspodijeljenog sustava osjetila temeljenih na mmWave tehnologiji koji bi omogućio učinkovito praćenje aktivnosti i stanja osoba u kućanstvu. Implementirati predloženo rješenje korištenjem mmWave osjetila i mikrokontrolera ESP32 te ispitati svojstva razvijenog sustava.

Rok za predaju rada: 14. veljače 2025.

Zahvaljujem se svom mentoru, izv. prof. dr. sc. Igoru Čavracu, na podršci i stručnom vodstvu tijekom izrade ovog rada. Njegovi savjeti i konstruktivni komentari bili su od velike pomoći za kvalitetnu realizaciju ovog rada.

Sadržaj

1. Uvod	4
2. Izazovi nadzora aktivnosti u pametnim kućanstvima	6
2.1. Tehnologije nadzora i njihove karakteristike	6
2.2. Tehnički izazovi implementacije	7
2.3. Društveni aspekti i prihvaćenost tehnologije	7
2.4. Zaštita privatnosti u pametnim kućanstvima	8
3. Vrste aktivnosti u kućanstvu i načini praćenja	9
3.1. Aktivnosti u kućanstvu	9
3.2. Sistematizacija aktivnosti	11
3.2.1. Mikro-aktivnosti	11
3.2.2. Aktivnosti na razini prostorije	12
3.2.3. Aktivnosti na razini kućanstva	14
4. Osnove mmWave radara i mogućnosti detekcije	16
4.1. Radar	16
4.2. mmWave radar	16
5. Osjetila korištena u razvoju sustava za praćenje aktivnosti	20
5.1. 24GHz mmWave osjetilo za praćenje prisutnosti i pokreta unutar prostorije	20
5.1.1. Standardni način rada (Scene Mode)	22
5.1.2. Underlying Open način rada	23
5.2. 60GHz mmWave osjetilo za detekciju spavanja, mjerenje otkucaja srca te ritma disanja	23

5.3. 60GHz mmWave osjetilo za detekciju pada	25
6. Ispitivanje karakteristika odabranih osjetila	29
6.1. Sklopovska i programska podrška	29
6.2. Konfiguracija osjetila i postupak snimanja podataka	30
6.2.1. Povezivanje osjetila i ESP32	30
6.2.2. Postupak snimanja podataka	31
6.2.3. Čišćenje i obrada podataka	32
6.3. Primjeri podataka i koda	32
6.3.1. Primjer programskog koda za ESP32	32
6.3.2. Primjer podataka prije i poslije čišćenja	34
6.4. Osjetilo za prisutnost i pokrete	36
6.4.1. Opis osjetila i scenariji testiranja	36
6.4.2. Zapažanja	37
6.5. Osjetilo za praćenje vitalnih znakova	39
6.5.1. Opis osjetila i scenariji testiranja	39
6.5.2. Zapažanja	41
6.6. Osjetilo za detekciju pada	45
6.6.1. Opis osjetila i scenariji testiranja	45
6.6.2. Zapažanja	45
6.7. Rezultati ispitivanja osjetila	45
7. Opis razvijenog sustava	47
7.1. Pregled arhitekture sustava	47
7.2. Tok podataka u sustavu	48
7.3. Mreža osjetila	48
7.4. Konfiguracija kućanstva	48
7.5. Centralno računalo	50
7.6. Simulator aktivnosti	57
7.7. Podrška za stvarna osjetila	65
7.8. Testiranje sustava	71
8. Zaključak	77

Literatura	79
Sažetak	82
Abstract	83

1. Uvod

Brzi napredak u komunikacijskim tehnologijama i nagli rast Interneta stvari omogućili su da se fizički svijet neprimjetno poveže s aktuatorima, osjetilima i drugim računalnim elementima, uz stalnu mrežnu povezanost. Ovaj kontinuirano povezani fizički svijet s računalnim elementima tvori *pametnu okolinu* [1]. Uz to, demografske promjene i starenje stanovništva rezultirale su time da sve veći broj starijih osoba živi samostalno, što dodatno naglašava potrebu za tehnološkim rješenjima koja mogu olakšati svakodnevicu i pravovremeno detektirati potencijalno opasne situacije.

U današnje vrijeme, sve izraženija potreba za praćenjem aktivnosti osoba u njihovim kućanstvima, osobito spomenute starije populacije, dovela je do razvoja naprednih sustava koji kombiniraju različita osjetila i komunikacijske tehnologije. Problem koji pritom nastaje leži u potrebi neinvazivnog i pouzdanog načina prikupljanja podataka o kretanju, obrascima ponašanja i vitalnim funkcijama korisnika, uz očuvanje njihove privatnosti i jednostavnost korištenja. Upravo je taj aspekt — učinkovito prepoznavanje aktivnosti i brza detekcija potencijalno opasnih situacija — temelj ove problematike.

Kao rješenje, u ovom radu predlaže se sustav baziran na mmWave osjetilima, koji omogućuju praćenje kretanja, prisutnosti i ključnih vitalnih znakova bez narušavanja privatnosti, a istovremeno pružaju dovoljno detaljnih informacija o korisnikovoj aktivnosti. MmWave tehnologija nudi visoku preciznost i može se uspješno koristiti u zatvorenom prostoru, ne oslanjajući se na osvjetljenje niti na prikupljanje vizualnih podataka.

Metoda rješavanja problema temelji se na kombiniranju više komponenata:

(1) mmWave osjetila koji detektiraju pokret i prisutnost, (2) osjetila za vitalne znakove, (3) centralnog računala (poslužitelja) za obradu podataka te (4) frontend sučelja za pregled i analizu u stvarnom vremenu. Razvijena je i podrška za simulaciju rada osjetila radi testiranja sustava bez potrebe za stalnim fizičkim okruženjem. Korištene su stvarne snimke očitavanja iz eksperimenata s mmWave osjetilima kako bi se generirali podaci koji vjerno oponašaju rad sustava u stvarnoj primjeni.

Predloženo rješenje karakterizira modularnost — moguće je jednostavno dodavati nove vrste osjetila ili proširivati broj postojećih, ovisno o potrebi dotičnog kućanstva. Sustav je dizajniran tako da bude prilagodljiv različitim rasporedima prostorija i razinama tehničkog znanja korisnika. Naglasak je stavljen na minimalnu invazivnost, pouzdanu detekciju aktivnosti te mogućnost integracije s postojećim ili budućim uređajima unutar koncepta pametnih kućanstava.

U nastavku rada, u drugom poglavlju analiziraju se izazovi nadzora aktivnosti u pametnim kućanstvima, s naglaskom na starije osobe i pitanja privatnosti. Treće poglavlje opisuje vrste aktivnosti u kućanstvu i načine njihova praćenja, uključujući klasifikaciju na mikro-aktivnosti, aktivnosti unutar prostorije i aktivnosti na razini kućanstva. Četvrto poglavlje donosi osnove mmWave radara i prikazuje glavne mogućnosti detekcije. U petom poglavlju predstavljena su osjetila korištena u razvoju sustava, dok se u šestom poglavlju opisuju pokusi i dobiveni rezultati ispitivanja njihovih performansi. Sedmo poglavlje donosi detaljan opis razvijenog sustava te arhitekture za obradu i vizualizaciju podataka. Konačno, osmo poglavlje zaključuje rad i iznosi smjernice za mogući budući razvoj.

2. Izazovi nadzora aktivnosti u pametnim kućanstvima

Pametna kućanstva sve se češće koriste za poboljšanje sigurnosti i kvalitete života, posebno među starijom populacijom koja živi samostalno. Napredne tehnologije osjetila omogućuju praćenje različitih aktivnosti korisnika, pružajući mogućnost pravovremene reakcije u slučaju potencijalno opasnih situacija. Takvi sustavi mogu detektirati obrasce kretanja, navike spavanja te prepoznati iznenadne promjene poput padova, što može biti ključno za zdravstvenu skrb i brzu intervenciju.

Međutim, unatoč njihovim prednostima, implementacija ovih tehnologija nosi i određene izazove. Ključni problemi odnose se na pouzdanost osjetilnih sustava, prihvaćenost tehnologije od strane korisnika te zaštitu privatnosti i osobnih podataka. Starije osobe često nisu navikle na korištenje digitalnih tehnologija, a osjećaj nadzora može izazvati nelagodu i nepovjerenje. Stoga je važno pronaći ravnotežu između povećane sigurnosti i očuvanja osjećaja privatnosti i autonomije korisnika.

U ovom poglavlju analiziraju se tehnički i društveni izazovi implementacije osjetilnih sustava u pametnim kućanstvima, s posebnim naglaskom na prihvaćenost ovih tehnologija među starijom populacijom te njihovu usklađenost s principima zaštite privatnosti.

2.1. Tehnologije nadzora i njihove karakteristike

Suvremena pametna kućanstva koriste raznovrsne tehnologije nadzora koje omogućavaju praćenje raznih aspekata svakodnevnog života. Osnova ovih sustava su

osjetila pokreta i prisutnosti, poput PIR osjetila te kontaktnih osjetila na vratima i prozorima koji prate ulazak i izlazak osobe iz prostorija.

Za točnije praćenje specifičnih aktivnosti, poput praćenja kvalitete sna ili mjerenje otkucaja srca te ritma disanja, sustavi često uključuju dodatne tehnologije. Primjerice, pametni kreveti opremljeni su osjetilima za praćenje kvalitete sna i odmora, dok su s druge strane primjerice pametna ogledala u kupaonici opremljena osjetilima za praćenje vitalnih znakova. Kao dodatni sloj sigurnosti korisnika, sve veći broj sustava uključuje i pametne narukvice i slične nosive predmete s pomoću koji omogućuju praćenje vitalnih znakova korisnika te mogu brzo reagirati na opasne situacije poput pada osobe.

2.2. Tehnički izazovi implementacije

Implementacija sustava nadzora nosi sa sobom niz složenih tehničkih izazova. Jedan od najtežih problema s kojima se treba suočiti je pouzdanost očitavanja s osjetila i minimizacija lažnih alarma, budući da ova dva problema zahtijevaju pravilnu kalibraciju i održavanje sustava. Različite tlocrtne konfiguracije kućanstava dodatno kompliciraju procese instalacije i optimizacije sustava, budući da se svako kućanstvo mora pojedinačno dizajnirati u smislu pozicije i postavki senzora.

Osnovni izazov dizajniranja jednog usklađenog sustava koji uključuje različita osjetila osobito je težak u pogledu prikupljanja i obrade podataka iz različitih izvora, kao i njihove inteligentne interpretacije. Također, kao i svi drugi moderni sustavi, trebaju biti energetske učinkoviti i biti sposobni za produljenu autonomiju, dok istovremeno trebaju moći primiti periodična ažuriranja koja omogućuju uključivanje novih funkcionalnosti i poboljšanje postojećih.

2.3. Društveni aspekti i prihvaćenost tehnologije

Korištenje bilo kojeg oblika tehnologije za praćenje nosi društvene posljedice, posebno kod starijih osoba. Velik broj starijih korisnika posjeduje određeni stupanj skepticizma prema novim tehnologijama, što se očituje kroz anksioznost prema tehnologiji zbog nedostatka iskustva s digitalnim uređajima. Ova pojava često je

popraćena zabrinutošću zbog gubitka osjećaja privatnosti i kontrole nad vlastitim životnim okruženjem.

Kako bi se učinkovito riješili ovi problemi, potrebno je razviti jednostavno i lako razumljivo sučelje te osigurati dovoljno edukacije i kontinuirane podrške. Potreba za poboljšanjem prihvatljivosti tehnologije postiže se postupnim uvođenjem novih funkcija uz aktivno sudjelovanje korisnika u odabiru i prilagodbi sustava. Tijekom ovog procesa ne smije se zaboraviti na posebne ili opće stavove korisnika prema različitim razinama tehnološke upućenosti u njihovom svakodnevnom životu.

2.4. Zaštita privatnosti u pametnim kućanstvima

Pitanje zaštite privatnosti predstavlja jedan od aspekata implementacije sustava nadzora u pametnim kućanstvima. Uspješno rješavanje ovog izazova zahtijeva razvoj transparentnih protokola za prikupljanje i obradu podataka, uz implementaciju sigurnih metoda pohrane i prijenosa informacija. Posebnu pažnju potrebno je posvetiti razvoju minimalno invazivnih metoda praćenja koje omogućuju očuvanje privatnosti korisnika.

Preporučeni pristup uključuje korištenje osjetila koja se aktiviraju samo na određene događaje umjesto kontinuiranog praćenja, čime se smanjuje količina prikupljenih podataka i povećava osjećaj kontrole kod korisnika. Gdje god je moguće, prednost se daje tehnologijama koje ne koriste vizualni nadzor, već se oslanjaju na alternativne metode prikupljanja podataka. Sustavi bi trebali omogućiti korisnicima jednostavnu kontrolu nad postavkama privatnosti i jasno razumijevanje načina na koji se njihovi podaci koriste.

Ovakvim sveobuhvatnim pristupom moguće je postići ravnotežu između potrebe za sigurnošću i očuvanja privatnosti korisnika, što predstavlja temelj za širu prihvaćenost ovih tehnologija u kontekstu samostalnog života starijih osoba. Uspješna implementacija zahtijeva kontinuiranu suradnju između tehnoloških stručnjaka, zdravstvenih djelatnika i samih korisnika, uz stalno praćenje i prilagodbu sustava prema povratnim informacijama iz prakse.

3. Vrste aktivnosti u kućanstvu i načini praćenja

3.1. Aktivnosti u kućanstvu

Aktivnosti svakodnevnog života (eng. Activities of Daily Living - ADL) je pojam koji se koristi za zajedničko opisivanje temeljnih vještina potrebnih za samostalnu brigu o sebi, poput jedenja, kupanja i kretanja. Pojam "ADL" prvi je put uveo Sidney Katz 1950. godine. [2].

Aktivnosti možemo podijeliti na osnovne i instrumentalne aktivnosti. Osnovne aktivnosti svakodnevnog života (Basic Activities of Daily Living - BADL) uključuju hranjenje, oblačenje, održavanje higijene i kretanje. Instrumentalne aktivnosti svakodnevnog života (Instrumental Activities of Daily Living - IADL) obuhvaćaju naprednije vještine poput upravljanja osobnim financijama, korištenja prijevoza, telefoniranja, kuhanja, obavljanja kućanskih poslova, pranja rublja i kupovine.[3]

Nadalje, aktivnosti se mogu klasificirati kroz taksonomiju u tri razine razlučivanja po stupnju apstrakcije počevši od nižeg prema višem stupnju:[4]

- Osnovno prepoznavanje pokreta koje se odnosi na tzv. *akcije* ili *motoričke primitive*, koji čine temeljne elemente od kojih se akcije sastoje. Može se reći da akcijski primitivi predstavljaju pokrete na razini udova.
- Skup različitih ili ponavljajućih akcijskih primitiva oblikuje jednu *radnju*.
- Na višoj razini, uzimajući u obzir širi kontekst, uključujući okolinu te interakciju s objektima ili drugim ljudima, moguće je prepoznati cjelokupnu **aktivnost**.

Pored ranije spomenutih klasifikacija aktivnosti, još jedan pristup kategorizaciji aktivnosti opisan je u radu [5], gdje su autori istražili mogućnost klasifikacije aktivnosti u pametnim domovima koristeći podatke prikupljene osjetilima. Njihova podjela temelji se na standardiziranim ljestvicama za procjenu autonomije starijih osoba, pri čemu su definirali sedam ključnih aktivnosti relevantnih za svakodnevni nadzor:

- Spavanje,
- Priprema i konzumacija obroka,
- Odijevanje i skidanje odjeće,
- Odmaranje,
- Higijenske aktivnosti,
- Korištenje toaleta,
- Komunikacija putem telefona.

Podaci prikupljeni s pomoću različitih osjetila (PIR osjetila prisutnosti, osjetila na vratima, mikrofoni i akcelerometri) korišteni su za automatsku klasifikaciju ovih aktivnosti. Ovakav pristup naglašava potrebu za raznolikim metodama detekcije koje uzimaju u obzir ne samo fizičke pokrete već i interakciju s okolinom.

Međutim, u sklopu ovog rada koristit će se nešto drugačija taksonomija od gore navedene. Kako je ciljna skupina korisnika sustava koji se razvija s ovim radom starija populacija tako je i raščlamba aktivnosti prilagođena takvim kućanstvima i vrstama aktivnosti koje se u istima odvijaju. Tako razlikujemo sljedeće vrste aktivnosti:

- **Mikro-aktivnosti** – odnose se na osnovne radnje koje korisnik obavlja u specifičnom trenutku, poput mirovanja, spavanja ili kretanja, koje se detektiraju osjetilima koji služe za detekciju promjene položaja tijela, disanje ili pokrete.
- **Aktivnosti unutar prostorije** – obuhvaćaju složenije radnje koje se de-

finiraju kombinacijom podataka iz osjetila za mikro-aktivnosti i osjetila za praćenje rada objekata s kojima korisnik ima interakciju u prostoriji, poput uključivanja televizora, kuhanja vode u kuhalu ili otvaranja hladnjaka.

- **Aktivnosti u kontekstu kućanstva** – uključuju integraciju aktivnosti iz više prostorija radi prepoznavanja šireg obrasca ponašanja u kućanstvu. Primjerice, detektira se da korisnik spava u jednoj prostoriji dok se istovremeno bilježi potrošnja električne energije na štednjaku, što može ukazivati na nesinkronizirane radnje ili potencijalno opasne situacije.

Ovakva raščlamba osigurava hijerarhijsko i logičko razlučivanje aktivnosti, od jednostavnih pokreta do složenih obrazaca koji obuhvaćaju cijelo kućanstvo.

3.2. Sistematizacija aktivnosti

Aktivnosti u kućanstvu razvrstavaju se na tri razine: mikro-aktivnosti, aktivnosti na razini prostorije i aktivnosti na razini kućanstva. Svaka od ovih kategorija ima svoje specifičnosti i načine detekcije, a u nastavku slijede primjeri za svaku od njih.

3.2.1. Mikro-aktivnosti

Mikro-aktivnosti predstavljaju osnovne radnje koje osoba obavlja u specifičnom trenutku, a detekcija tih aktivnosti zavisi od udaljenosti, položaju tijela u odnosu na osjetilo te uvjetima okoline. Primjeri mikro-aktivnosti uključuju osnovne fiziološke funkcije i promjene položaja tijela.

Mikro-aktivnosti obuhvaćaju:

- **Mirovanje** – stanje u kojem korisnik ostaje u jednom položaju, obično u sjedećem ili ležećem položaju, bez značajnijih pokreta.
- **Spavanje** – aktivnost koja se obično prepoznaje na temelju promjena u ritmu disanja i srčanog ritma.
- **Disanje (ritam disanja)** – praćenje ritma disanja korisnika.

- **Srčani ritam** – praćenje otkucaja srca, često korišteno za ocjenu fizičkog stanja korisnika.
- **Prisutnost u prostoriji** – detekcija prisutnosti osobe u određenoj prostoriji na temelju podataka dobivenih s osjetila.
- **Pad** – detekcija fizičkog pada korisnika, što može zahtijevati visoku osjetljivost i preciznost osjetila.
- **Razlikovanje mirovanja i spavanja** – u nekim slučajevima, mirovanje može biti zamijenjeno sa spavanjem, što zahtijeva dodatnu analizu ritma disanja i drugih fizioloških parametara.
- **Mirovanje uz neuobičajen ritam disanja ili rada srca** – prepoznavanje kada korisnik miruje, ali uz abnormalne fiziološke znakove koji mogu ukazivati na zdravstvene probleme.

3.2.2. Aktivnosti na razini prostorije

Aktivnosti na razini prostorije obuhvaćaju složenije radnje koje uključuju interakciju osobe s objektima u prostoru, uz praćenje njihovih pokreta i promjena u okruženju. Ove aktivnosti obično zahtijevaju kombinaciju podataka s više osjetila, kao što su osjetila pokreta, osjetila za prepoznavanje objekata, te osjetila za praćenje korištenja kućanskih aparata.

Dnevna soba

U dnevnoj sobi, aktivnosti mogu biti jednostavne, poput čitanja ili gledanja TV-a, ili zahtijevati više fizičkog angažmana, poput ustajanja i sjedenja. Na primjer:

- **Sjedenje – čitanje i sjedenje – pletenje** predstavljaju mirne aktivnosti koje uključuju sjedenje, ali uz povremene manje pokrete.
- **Ležanje na kauču – gledanje TV-a / čitanje knjige** predstavlja ležeću aktivnost koja uključuje povremene pokrete (npr. okretanje stranica ili upotreba daljinskog upravljača).

- **Kretanje (ulazak/izlazak iz prostorije)** obuhvaća fizičko kretanje osobe izvan prostorije, što može ukazivati na promjenu aktivnosti ili prelazak u drugu prostoriju.
- **Ležanje na podu – mirovanje na neuobičajenom mjestu** može ukazivati na nezgodu ili nenormalno ponašanje koje zahtijeva pažnju.
- **Pad** može se dogoditi u bilo kojem trenutku, a osjetila za detekciju padova mogu pomoći u pravovremenom prepoznavanju i reakciji.

Kuhinja

U kuhinji aktivnosti uključuju i sjedeće radnje, poput čitanja ili objedovanja, ali i stajace aktivnosti koje mogu uključivati kuhanje ili pranje suđa:

- **Sjedenje za stolom – čitanje** ili **sjedenje za stolom – objed** odnose se na aktivnosti koje uključuju sjedenje s povremenim manjim pokretima.
- **Stajace aktivnosti u kuhinji** uključuju kuhanje ili pranje suđa, gdje su pokreti tijela češći i dinamičniji.
- **Ležanje na podu – mirovanje na neuobičajenom mjestu** može ukazivati na potencijalni pad ili nezgodu u kuhinji.
- **Pad** je, kao i u dnevnoj sobi, ključna aktivnost koja se mora pratiti kako bi se osigurala sigurnost korisnika.

Spavaća soba

Aktivnosti u spavaćoj sobi uglavnom uključuju mirovanje i spavanje, no s obzirom na zdravstveno stanje korisnika, mogu biti uključene i druge radnje:

- **Ležanje na krevetu – čitanje** i **ležanje na krevetu – mirovanje (nije spavanje)** mogu ukazivati na mirne aktivnosti, dok **ležanje na krevetu – spavanje** označava standardnu aktivnost spavanja.
- **Ležanje na krevetu – sumnjiv status** može ukazivati na neobičnu aktivnost koja zahtijeva dodatnu pažnju (npr. nepravilni pokreti ili promjena

stanja).

- **Kretanje (ulazak/izlazak iz prostorije)** pokazuje aktivnost korisnika u prostoru, uključujući odlazak na spavanje ili ustajanje.
- **Ležanje na podu – mirovanje na neuobičajenom mjestu** također može ukazivati na nezgodu.
- **Pad** mora biti prepoznat u bilo kojoj situaciji.

Hodnik, Kupaonica, WC

U hodniku, kupaonici i WC-u, aktivnosti se uglavnom odnose na kretanje između prostorija, oblačenje, te potencijalne nezgode:

- **Kretanje (ulazak/izlazak iz prostorije)** označava osnovnu aktivnost premještanja između prostorija.
- **Ležanje na podu – mirovanje na neuobičajenom mjestu** može ukazivati na opasne situacije poput pada.
- **Pad** je uvijek kritičan u ovim prostorijama, jer padovi mogu biti ozbiljni u uskim prostorijama poput WC-a ili kupaonice.

3.2.3. Aktivnosti na razini kućanstva

Na razini kućanstva, aktivnosti obuhvaćaju šire obrasce ponašanja koji uključuju više prostorija i kombiniraju različite aktivnosti korisnika. Ovdje se prate interakcije između prostorija i mogu se prepoznati nesinkronizirani obrasci aktivnosti:

- **Kretanje između prostorija** obuhvaća sve aktivnosti vezane uz premještanje korisnika iz jedne prostorije u drugu, što može biti ključ za prepoznavanje šireg obrasca ponašanja.
- **Praćenje stvarne prostorije u kojoj je osoba** odnosi se na točnu detekciju gdje se osoba nalazi u određenom trenutku, što je od velike važnosti za pravilno praćenje aktivnosti.

- **Ulaz/izlaz iz stana** može ukazivati na ključne promjene u obrascima ponašanja, poput napuštanja ili dolaska u stan.
- **Sinkronizacija aktivnosti** je važna za prepoznavanje situacija kada korisnik spava ili je neaktivan, dok se u drugoj prostoriji detektira aktivnost, poput rada štednjaka.

Kako se radi o ciljanoj skupini korisnika sustava razvijenog uz ovaj rad, tako su i osjetila odnosno uređaji koji će detektirati i pratiti ove aktivnosti prilagođeni njihovim specifičnim potrebama i izazovima. Naime, koristit će se mmWave radari, koji, zahvaljujući svojoj neinvazivnoj prirodi i sposobnosti praćenja aktivnosti bez narušavanja privatnosti, predstavljaju dobro rješenje za stariju populaciju. Sljedeće poglavlje detaljnije će predstaviti karakteristike i prednosti mmWave osjetila te objasniti njihovu primjenu u kontekstu razvoja sustava za pametna kućanstva.

4. Osnove mmWave radara i mogućnosti detekcije

Ovo poglavlje se fokusira na mmWave (eng. *millimeter-wave*) radare, jer su to osjetila koja su se koristila u ovom radu i sustava koji se razvija s njim. Kroz ovo poglavlje ukratko će se objasniti princip rada mmWave radara, uspoređujući pritom neke karakteristike s klasičnim radarom te će se dati usporedba s ostalim osjetilima koji koriste drugačije načine za dobivanje podataka iz okoline, odnosno koriste drugačije tehnologije.

4.1. Radar

Radar (eng. *Radio Detection and Ranging*) je uređaj koji koristi elektromagnetske valove za detekciju objekata i određivanje njihove udaljenosti, brzine i smjera. Osnovno načelo rada radara temelji se na odašiljanju radio valova iz antene te detekciji reflektiranih valova od objekata u okruženju. U tu svrhu upotrebljavaju se decimetarski ili centimetarski valovi (valovi valne duljine u spomenutim redovima veličine) [6]. Vrijeme koje je potrebno valovima da stignu do objekta i vrate se naziva se vrijeme leta (*time-of-flight*). Iz ovog vremenskog intervala moguće je odrediti udaljenost objekta, dok analiza promjene frekvencije reflektiranog signala (Dopplerov efekt) omogućuje mjerenje brzine objekta.

4.2. mmWave radar

Za razliku od klasičnog radara, mmWave radar je specijalna klasa radara koja koristi elektromagnetske valove puno kraće valne duljine te istodobno u puno većem frekvencijskom rasponu [7]. Primjerice, većina nautičkih radara radi u frekvencijskom pojasu od 10GHz (3 cm valne duljine) i 3 GHz (10 cm valne du-

ljine). Takvi radari su dobri za detekciju objekata na udaljenostima od 20 nautičkih milja (cca. 37 km) [8]. Za usporedbu, mmWave radari najčešće djeluju u frekvencijskom rasponu od 30-300GHz (pri čemu raspon valne duljine varira od 1 mm do 10 mm) [9].

Dodatno, za razliku od klasičnog radara koji koristi vrijeme potrebno da se odašiljani impuls vrati nakon refleksije s objektom, mmWave radari funkcioniraju na principu FMCW (eng. *Frequency modulated continuous wave*) tehnologije; analiziraju frekvencijsku razliku u fazi frekvencije između odašiljanog i primljenog signala kako bi odredili udaljenost do objekta kojeg detektiraju [10]. Osnovne funkcionalnosti FMCW radara su:

- sposobnost mjerenja udaljenosti i relativne brzine objekta
- sposobnost mjerenja vrlo malih udaljenosti objekta od samog radara (zahvaljujući tome što se gleda fazni pomak odašiljanog i reflektiranog signala)
- visoka preciznost mjerenja udaljenosti do objekta
- procesiranje signala obavlja se u niskom rasponu frekvencija, što znatno pojednostavljuje implementaciju sklopovlja uređaja
- koristi kontinuirani signal s niskom snagom, bez visokih vršnih impulsa, čime smanjuje rizik za ljude i opremu te omogućuje sigurnu primjenu u osjetljivim okruženjima

Spektar primjene ove vrste radara vrlo je širok pa se tako koriste u raznim područjima i industrijama. Neka od područja primjene su:

- **Automobilska industrija:** koriste se za napredne sustave potpore u vožnji (eng. *ADAS*), detekciju sudara i pomoć pri parkiranju
- **Robotika:** koriste se detekciju prepreka, navigaciju, vizualizaciju okoline te za robotske osjetilne sustave
- **Meteorološke primjene:** pomažu u analizi oblaka, analizi usjeva, radiometriji, geografsko informacijskim sustavima tzv. GIS i proučavanju klime

- **Zrakoplovstvo i vojni sustavi:** za presretanje i navođenje projektila te bespilotnih letjelica
- **Upotreba u industrijskim i automatiziranim sustavima:** koriste se za kontrolu kvalitete, detekciju pukotina, mjerenje tekućina i volumena, otkrivanje curenja i klasifikaciju materijala
- **Medicina:** detekcija i praćenje ljudskih pokreta i gesta, mjerenje otkucaja srca te ritma disanja
- **Primjene u društvenim i javnim uslugama:** koriste se u svrhu kontrole zračnog prometa, nadzora na niskim visinama, u sustavima ranog upozorenja za izbjegavanje rizika, pametnim gradovima te pametnim kućanstvima [11]

U kontekstu praćenja aktivnosti u pametnom kućanstvu, mmWave radarska osjetila odlikuju mnoge prednosti u odnosu na konkurentne tehnologije praćenja aktivnosti poput kamera i PIR (eng. *passive infrared*) osjetila. U usporedbi s kamerama koje koriste stvarni prikaz prostora i osoba kako bi obrađivale podatke, što znatno utječe na privatnost korisnika, mmWave osjetila ne narušavaju privatnost korisnika jer ne obrađuju vizualne podatke. Također kamere zahtijevaju strože uvjete od onih koji su potrebni da bi mmWave osjetila uspješno detektirala aktivnost; kamere zahtijevaju određeni stupanj osvjetljenja prostorije u kojoj se vrši detekcija dok mmWave osjetila ne trebaju svjetlost unutar prostorije kako bi otkrili aktivnost. Što se tiče PIR osjetila, kao i mmWave osjetila ne zahtijevaju osvjetljenost prostora kako bi ispravno funkcionirali i detektirali aktivnosti (u slučaju PIR osjetila pokret). No ono što uvelike razlikuje mmWave osjetila od PIR osjetila jest činjenica da PIR osjetila mogu detektirati samo prisutnost objekta na temelju toplinskog potpisa osobe, a mmWave osjetila mogu detektirati prisutnost, relativnu brzinu kretanja osobe, potencijalne padove te mjeriti vitalne funkcije poput otkucaja srca te ritma disanja. Također, mmWave osjetila nisu osjetljiva na veće promjene temperature koje na PIR osjetila mogu utjecati na očitavanja s obzirom na to da im je detekcija objekta zasnovana na promjeni iste [12].

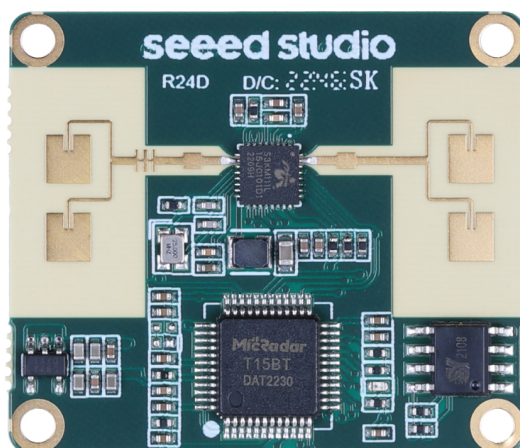
Unatoč prednostima, mmWave radari imaju i svoja ograničenja. Na primjer, njihov raspon detekcije može biti manji u odnosu na klasične radare koji rade na nižim frekvencijama, jer više frekvencije imaju veću apsorpciju signala od strane prepreka poput zidova ili vlage, što može smanjiti učinkovitost rada u određenim uvjetima.

5. Osjetila korištena u razvoju sustava za praćenje aktivnosti

U svrhu stvaranja sustava za praćenje aktivnosti u kućanstvu koji ovaj rad opisuje odabrani su sljedeća osjetila: 24GHz mmWave osjetilo za praćenje prisutnosti i kretanja unutar prostorije, 60GHz mmWave osjetilo za određivanje otkucaja srca te ritma disanja te 60GHz mmWave osjetilo za otkrivanje pada. U nastavku ovog poglavlja bit će opisani rad pojedinih osjetila i dana usporedba njihovih karakteristika.

5.1. 24GHz mmWave osjetilo za praćenje prisutnosti i pokreta unutar prostorije [13, 14, 15]

24GHz mmWave osjetilo za praćenje prisutnosti i pokreta koristi tehnologiju milimetarskih valova za detekciju statičkih i dinamičkih aktivnosti unutar prostorije. Prikaz osjetila moguće je vidjeti na slici 5.1. Ovo osjetilo emitira elektromagnetske valove koji se reflektiraju od objekata i osoba unutar dometa. Analizom reflektiranih signala, osjetilo procjenjuje ključne parametre kao što su prisutnost, udaljenost, brzina i intenzitet pokreta. Njegov rad temelji se na mikro-pokretima (poput disanja ili vrlo blagih pokreta) i aktivnim kretanjima, što ga čini posebno korisnim za nadzor aktivnosti u zatvorenim prostorima. Osim toga, osjetilo reagira na promjene u okolišu te može identificirati čak i suptilne promjene koje ukazuju na prisutnost osobe, uključujući osobe koje miruju. U opsegu ovog rada koristit će se u svrhu detekcije aktivnosti iz svih prethodno opisanih kategorija aktivnosti unutar kućanstva (mikro-aktivnosti, aktivnosti na razini pojedine prostorije te aktivnosti na razini kućanstva).



Slika 5.1. 24GHz mmWave osjetilo za prisutnost i pokrete, pogled sprijeda

Osjetilo je unutar prostorije moguće instalirati horizontalno, dijagonalno odnosno na stropu neke prostorije nagnut pod kutom ($55-60^\circ$ s obzirom na strop) gdje se koristi za detekciju pokreta i prisutnosti unutar prostorije te na kraju može se montirati vertikalno na stropu. U nastavku rada ovo osjetilo koristiti će se montirano na strop.

Ovo osjetilo podržava dva različita načina rada koji omogućuju prilagodbu ovisno o potrebama razvijatelja sustava te korisnika istog. Prvi je **standardni način rada (Scene Mode)**, koji je osmišljen za jednostavne aplikacije i koristi unaprijed definirane funkcionalnosti poput detekcije prisutnosti, pokreta i mikro-pokreta. Ovaj način rada nudi predefinirane scenarije (dnevna soba, spavaća soba, kupaonica) i omogućuje osnovne prilagodbe poput razine osjetljivosti. Drugi je **underlying open način rada**, koji pruža detaljne niskorazinske podatke, poput energije pokreta, udaljenosti i brzine. Ovaj način rada namijenjen je naprednijim korisnicima koji žele veću kontrolu i prilagodbu osjetila za specifične aplikacije, kao što su napredna analitika pokreta ili filtriranje lažnih alarma. Opis oba načina rada je u nastavku.

5.1.1. Standardni način rada (Scene Mode)

Standardni način rada ovog osjetila osmišljen je za jednostavne aplikacije s unaprijed definiranim funkcionalnostima. Ovaj način rada omogućuje detekciju prisutnosti, pokreta, mikro-pokreta te stanja poput spavanja. Osjetilo pruža predefinirane scenarije za različite prostorije:

- Dnevna soba: doomet detekcije 4–4,5 m,
- Spavaća soba: doomet detekcije 3,5–4 m,
- Kupaonica: doomet detekcije 2,5–3 m,
- Zona detekcije: doomet detekcije 3–3,5 m.

Komunikacija se odvija putem UART protokola sa sljedećim specifikacijama: brzina prijenosa 9600 bps, 8-bitni podaci, 1 stop bit i ne koristi se paritet kao zaštita od mogućih grešaka pri slanju poruka. Svaka poruka sastoji se od fiksne strukture:

- **Zaglavlje okvira:** 2 bajta (0x53, 0x59),
- **Kontrolna riječ:** 1 bajt (npr. 0x80 za funkcije prisutnosti),
- **Naredbena riječ:** 1 bajt (npr. 0x01 za upit prisutnosti),
- **Dužina podataka:** 2 bajta,
- **Podaci:** varijabilne duljine, ovisno o funkciji,
- **Kontrolni zbroj:** 1 bajt (suma svih prethodnih polja, niži bajt),
- **Kraj okvira:** 2 bajta (0x54, 0x43).

Primjer upita za prisutnost:

0x53, 0x59, 0x80, 0x81, 0x00, 0x01, 0x0F, 0x54, 0x43

Odgovor osjetila uključuje informaciju o prisutnosti (0x01 za prisutnost, 0x00 za odsutnost).

5.1.2. Underlying Open način rada

Underlying open način rada pruža pristup detaljnim niskorazinskim informacijama, omogućujući naprednu analizu i prilagodbu. Osjetilo izvještava o sljedećim parametrima:

- Energija postojanja (eng. *existence energy*): raspon 0–250,
- Udaljenost od osjetila (eng. *static distance*): 0,5–3 m,
- Energija pokreta (eng. *motion energy*): raspon 0–250,
- Udaljenost pokreta (eng. *motion distance*): 0,5–4 m,
- Brzina pokreta (eng. *motion speed*): raspon ± 5 m/s.

Komunikacija koristi istu strukturu okvira kao i standardni način rada, ali s proširenim opcijama za kontrolne i zapovjedne riječi. Primjer izvještaja o energiji pokreta:

0x53, 0x59, 0x08, 0x82, 0x00, 0x01, 0x0F, 0x54, 0x43

Odgovor uključuje vrijednost energije pokreta (u ovom slučaju 0x1E označava vrijednost 30). Ovaj način rada omogućuje postavljanje prilagodljivih pragova i granica, čime se kontrolira preciznost detekcije i smanjuje broj lažnih alarma.

5.2. 60GHz mmWave osjetilo za detekciju spavanja, mjerenje otkucaja srca te ritma disanja [16, 17, 18]

60GHz mmWave osjetilo koristi tehnologiju milimetarskih valova za neinvazivno praćenje otkucaja srca te ritma disanja. Prikaz osjetila moguće je vidjeti na slici 5.2. Ovo se osjetilo također, kao i prethodno, temelji na FMCW načinu rada, što omogućava precizno praćenje vitalnih znakova osobe na udaljenosti od 0,4 do 2 metra. Mjerenje se odvija kroz praćenje pokreta na površini velikih mišića koristeći frekvenciju emitiranih valova od 60GHz. Kako osoba diše, tako osjetilo s pomoću pokreta koji se odvijaju na području prsa i leđa uspijeva mjeriti ritam disanja te otkucaje srca. Ovo je osjetilo također moguće instalirati na više načina.

Prvi način je vertikalno iznad osobe kojoj se mjere vitalni znakovi, sljedeći je vertikalno bočno u slučaju da strop iznad subjekta nije ravan te ga je moguće instalirati na zid ili neko postolje tako da osjetilo ima "pogled" na torzo osobe koja sjedi ili stoji ispred osjetila. U sklopu ovog rada koristit će se za mjerenje vitalnih znakova korisnika sustava iako se isti može koristiti i za detekciju prisutnosti te tako moguće smanjiti broj osjetila koji se instaliraju unutar prostorije u kojoj se obavlja mjerenje vitalnih znakova.



Slika 5.2. 60GHz mmWave osjetilo za mjerenje otkucaja srca i ritma disanja

Ovo osjetilo podržava višestruke funkcionalnosti i režime rada za detekciju specifičnih parametara:

- **Praćenje ritma disanja i otkucaja srca:**

- Osjetilo bilježi i analizira amplitude reflektiranih valova kako bi izračunao frekvencije otkucaja srca i respiratornih pokreta.
- Podaci se osvježavaju svake 3 sekunde.

- **Prepoznavanje prisutnosti i kretanja:**

- Osjetilo detektira razlike između statičkih i dinamičkih objekata na udaljenosti do 3 m.

- **Praćenje spavanja:**

- Kategorije stanja: budan, lagan san, dubok san.
- Prikaz kvalitete spavanja nakon završetka procesa spavanja.

Komunikacija se odvija putem UART protokola sa sljedećim specifikacijama: brzina prijenosa 115200 bps, 8-bitni podaci, 1 stop bit i ne koristi se paritet kao zaštita od mogućih grešaka pri slanju poruka. Svaka poruka sastoji se od fiksne strukture:

- **Zaglavlje:** 0x53, 0x59.
- **Kontrolna riječ:** 1 bajt (npr. 0x85 za mjerenje otkucaja srca).
- **Naredbena riječ:** 1 bajt.
- **Duljina podataka:** 2 bajta.
- **Podaci:** varijabilne duljine.
- **Kontrolni zbroj:** 1 bajt.
- **Kraj okvira:** 0x54, 0x43.

Primjer poruke za upit otkucaja srca:

0x53, 0x59, 0x85, 0x02, 0x00, 0x01, 0x4B, 0x54, 0x43

Odgovor osjetila uključuje podatke o otkucajima srca (npr. 0x4B predstavlja mjerenje od 75 BPM).

5.3. 60GHz mmWave osjetilo za detekciju pada [19, 20, 21]

60GHz mmWave osjetilo za detekciju pada koristi milimetarske valove za praćenje prisutnosti i aktivnosti te detekciju kritičnih događaja poput pada. Prikaz osjetila moguće je vidjeti na slici 5.3. Ovo osjetilo funkcionira na sličnom principu kao i prethodno opisana osjetila u ovom poglavlju. Detekcija je prilagođena

za otkrivanje pada i drugih relevantnih događaja kao što su nepravilna kretanja ili dugotrajna nepomičnost. U opsegu ovog rada osjetilo će se koristiti za otkrivanje pada kod korisnika unutar prostorija u kojima se to češće događa poput kupaonice. Osim te funkcionalnosti, također se može koristiti za otkrivanje prisutnosti te se može razlikovati stacionaran položaj osobe (ležanje i mirno sjedenje, na primjer čitanje) te kretanje osobe unutar prostorije u kojoj je osjetilo postavljeno.



Slika 5.3. 60GHz mmWave osjetilo za detekciju pada

Osjetilo se može instalirati na različite načine, ovisno o primjeni i konfiguraciji prostora. Primaran način postavljanja jest montiranje na strop, ali moguće ga je postaviti i na zid ili specifična postolja, pri čemu je potrebno osigurati optimalnu pokrivenost prostora i smanjenje interferencija.

60GHz mmWave osjetilo za detekciju pada podržava višestruke funkcionalnosti i režime rada za detekciju kritičnih događaja:

- **Detekcija pada:**

- Osjetilo automatski izvještava o stanju pada kada se zadovolje uvjeti za alarm.

- Izvještaj uključuje razinu sigurnosti detekcije (eng. *confidence level*) i koordinate mjesta pada.
- **Nepokretnost osobe:**
 - Alarm se aktivira ako osoba ostane nepomična određeno vrijeme (prema unaprijed postavljenim pragovima).
- **Prilagodba osjetljivosti:**
 - Različiti pragovi osjetljivosti omogućuju prilagodbu detekcije specifičnim uvjetima okruženja.
- **Praćenje aktivnosti:**
 - Uključuje prepoznavanje prisutnosti i osnovnih pokreta kako bi se poboljšala preciznost detekcije.

Komunikacija se odvija putem UART protokola sa sljedećim specifikacijama: brzina prijenosa 115200 bps, 8-bitni podaci, 1 stop bit i ne koristi se paritet kao zaštita od mogućih grešaka pri slanju poruka. Svaka poruka sastoji se od fiksne strukture:

- **Zaglavlje:** 0x53, 0x59.
- **Kontrolna riječ:** 1 bajt (0x83 za funkcije detekcije pada).
- **Naredbena riječ:** 1 bajt.
- **Duljina podataka:** 2 bajta.
- **Podaci:** varijabilne duljine.
- **Kontrolni zbroj:** 1 bajt.
- **Kraj okvira:** 0x54, 0x43.

Primjeri poruka:

- Primjer upita za status pada:

0x53, 0x59, 0x83, 0x81, 0x00, 0x01, 0x0F, 0x54, 0x43

- Odgovor osjetila - Pad detektiran:

0x53, 0x59, 0x83, 0x81, 0x00, 0x01, 0x01, 0x54, 0x43

- Odgovor osjetila - Pad nije detektiran:

0x53, 0x59, 0x83, 0x81, 0x00, 0x01, 0x00, 0x54, 0x43

Sljedeća tablica 5.1. prikazuje usporedbu mogućnosti pojedinih osjetila. U razvijanju sustava bitno je ekonomično raspodijeliti osjetila kako ih se ne bi koristilo više no što treba, a istovremeno da se zadrže one mogućnosti detekcije koje su nužne za pojedine prostorije.

Mogućnost detekcije	24GHz mmWave osjetilo za detekciju prisutnosti i pokreta	60GHz mmWave osjetilo za praćenje vitalnih znakova	60GHz mmWave osjetilo za detekciju pada
Domet detekcije	5 m	2 m	6 m
Prisutnost	✓	✓	✓
Mirovanje	✓	✓	✓
Pokret	✓	✓	✓
Brzina pokreta	✓	-	-
Mjerenje udaljenosti	✓	-	-
Mjerenje blizine	✓	✓	✓
Detekcija pada	-	-	✓
Mjerenje otkucaja srca	-	✓	-
Mjerenje ritma disanja	-	✓	-

Tablica 5.1. Tablica usporedbe odabranih osjetila prema njihovim mogućnostima detekcije.

6. Ispitivanje karakteristika odabranih osjetila

Kako bi se ispitala funkcionalnost i karakteristike odabranih osjetila, provedena su ispitivanja u kontroliranim uvjetima, simulirajući stvarne scenarije iz kućanstva. Testiranja su omogućila prikupljanje i analizu podataka radi prepoznavanja obrazaca u mjerenjima te konzistentnosti i ograničenja svakog osjetila. U nastavku su opisani korišteni alati, metode ispitivanja i zapažanja.

6.1. Sklopovska i programska podrška

Za ispitivanje karakteristika odabranih osjetila korišteni su sljedeći alati:

- **ESP32 WROOM Devkit V1** – Mikrokontroler korišten za upravljanje osjetilima i prijenos podataka putem serijske komunikacije.
- **Arduino IDE** ¹ – Razvojno okruženje za pisanje i učitavanje programske podrške na ESP32. Korištene su knjižnice proizvođača osjetila preuzete s GitHub-a.
- **CoolTerm** ² – Alat za praćenje serijske komunikacije između računala i ESP32, korišten za spremanje podataka u .txt formatu.
- **Python** – Skripta za čišćenje i pretvorbu .txt datoteka u .csv format, kako bi podaci bili jednostavniji za analizu i obradu.

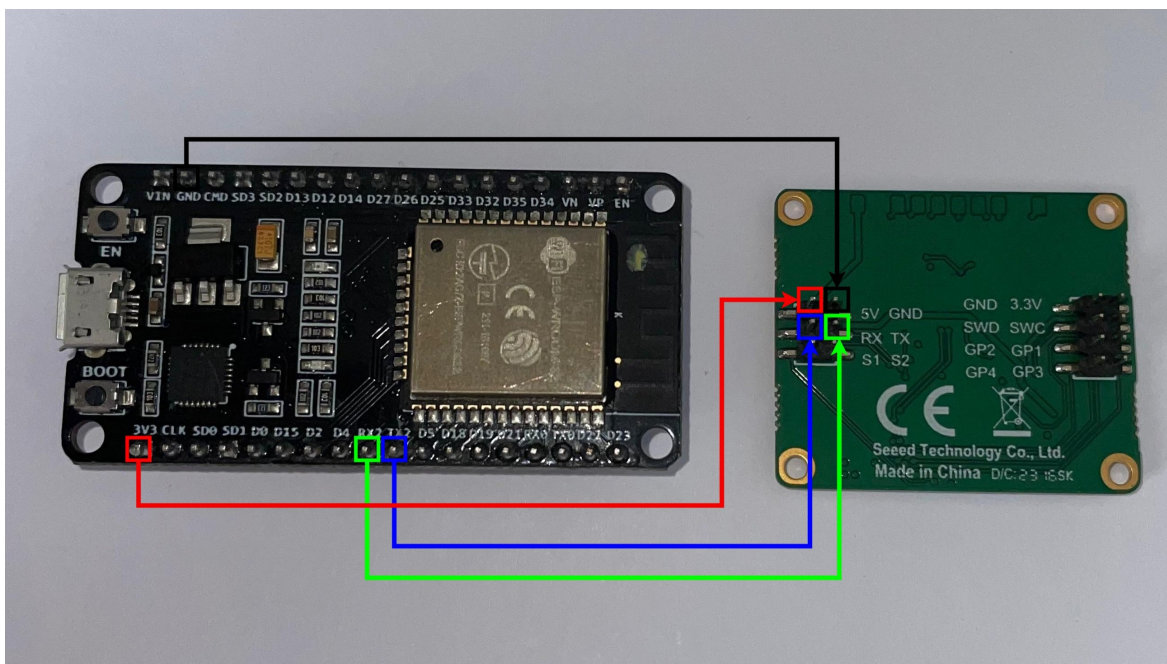
¹Arduino IDE razvojno okruženje preuzeto je sa sljedeće poveznice: <https://www.arduino.cc/en/software>

²Alat za praćenje serijske komunikacije, preuzet sa sljedeće web lokacije: <https://coolterm.sourceforge.com/windows>

6.2. Konfiguracija osjetila i postupak snimanja podataka

6.2.1. Povezivanje osjetila i ESP32

Serijska komunikacija između ESP32 i osjetila ostvarena je s pomoću **RX** i **TX** pinova mikrokontrolera. Na korištenom ESP32 WROOM Devkit V1 uređaju, korišteni su GPIO pinovi **16 (RX)** i **17 (TX)** za prijem i slanje podataka s osjetila. Komunikacija je uspostavljena s baud rate vrijednošću od **115200 bps** kako bi se osigurala stabilna i pouzdana razmjena podataka. Shema spajanja za ovaj specifični ESP32 mikrokontroler i osjetila vidljiva je na sljedećoj slici 6.1. Kako sva tri odabrana osjetila imaju veoma sličan, gotovo identičan raspored spojnih pinova, ova shema vrijedi za sva osjetila korištena u ovom radu.



Slika 6.1. Prikaz sheme spajanja osjetila i ESP32 mikrokontrolera

U Arduino IDE-u korištene su knjižnice proizvođača osjetila, što je omogućilo jednostavno dohvaćanje podataka putem funkcija iz tih knjižnica ^{3 4 5}. Mjere-

³Biblioteka potrebna za komunikaciju s osjetilom prisutnosti i pokreta preuzeta je s GitHub repozitorija na ovoj poveznici: https://github.com/limengdu/Seeed_24GHz_Human_Static_Presence_Module_Lite/tree/main

⁴Biblioteka potrebna za komunikaciju s osjetilom za vitalne znakove preuzeta je sa GitHub repozitorija na sljedećoj web lokaciji: <https://github.com/limengdu/Seeed-Studio-MR60BHA1-Sensor/tree/main>

⁵Biblioteka za komunikaciju s osjetilom detekcije pada preuzeta je sa sljedećeg GitHub repozitorija: <https://github.com/limengdu/Seeed-Studio-MR60FDA1-Sensor/tree/main>

nja su uključivala i vremenske oznake izražene u milisekundama, što omogućuje precizno praćenje trenutka u kojem su očitavanja zaprimljena.

Sva osjetila konfigurirana su na isti način kako bi se osigurala uniformnost među mjerenjima te jednostavnija analiza podataka. Struktura zaglavlja očišćenih podataka u CSV formatu:

- Za osjetilo prisutnosti: `time`, `status`, `bodysign_val`
- Za osjetilo vitalnih znakova: `time`, `heart_rate`, `breath_rate`

6.2.2. Postupak snimanja podataka

Svaka aktivnost snimana je iterativno – izvođena je više puta kako bi se osigurala konzistentnost podataka. Prije svake nove iteracije ili promjene aktivnosti, ESP32 je **resetiran** tako da ga se ugasio i ponovno uključilo, čime su se osigurali isključivo svježiji podaci i spriječilo miješanje očitavanja između različitih aktivnosti.

Na početku snimanja, programska podrška na ESP32 čekala je **10 sekundi** kako bi omogućila korisniku da se pripremi za izvođenje aktivnosti. U trenucima odbrojavanja, započeto je i samo snimanje podataka u tekstualnu datoteku kroz sučelje CoolTerm alata. Po završetku simulacije aktivnosti, snimanje je zaustavljeno ručno unutar CoolTerm sučelja, čime su završni podaci precizno obuhvaćeni bez neželjenih prijelaza između iteracija.

Podaci su prikupljeni s pomoću **CoolTerm-a**, gdje je uspostavljena serijska veza s ESP32 i svi dolazni podaci zapisivani su u `.txt` datoteku. Ove datoteke sadržavale su:

- Početne informacije poput odbrojavanja do početka snimanja.
- Heksadecimalne vrijednosti koje nisu interpretirane kroz programsku podršku na ESP32 mikrokontroleru, ali su ispisivane zbog same definicije knjižnice za odabrana osjetila.
- Formatirane podatke u skladu s definiranim strukturama za svako osjetilo.

6.2.3. Čišćenje i obrada podataka

Kako bi se osigurala kvalitetna obrada podataka, razvijena je **Python skripta** koja automatski filtrira sve .txt datoteke unutar zadanog direktorija i stvara novi direktorij s datotekama pročišćenog sadržaja u .csv formatu. Skripta funkcionira na slijedeći način:

1. **Uklanja nepotrebne linije** (poput odbrojavanja i heksadecimalnih podataka).
2. **Filtrira samo linije koje sadrže zarez** (što označava formatirana mjerenja).
3. **Sprema podatke u .csv formatu** radi lakšeg rukovanja i analize.

6.3. Primjeri podataka i koda

6.3.1. Primjer programskog koda za ESP32

```
1
2 #include "Arduino.h"
3 #include <60ghzbreathheart.h>
4
5 // Using HardwareSerial
6 BreathHeart_60GHz radar = BreathHeart_60GHz(&Serial1);
7
8 void setup() {
9     // Initialize Serial communication
10    Serial.begin(115200);
11    Serial1.begin(115200, SERIAL_8N1, 16, 17);
12
13    while (!Serial); // Wait for Serial Monitor connection
14
15    // Print a message indicating the 10-second delay
16    Serial.println("Starting in 10 seconds...");
17
18    // Wait for 10 seconds, printing the countdown every second
```

```

19  for (int i = 10; i > 0; i--) {
20      Serial.print("Starting in ");
21      Serial.print(i);
22      Serial.println(" seconds...");
23      delay(1000); // Wait 1 second
24  }
25
26  // Print the CSV header after the delay
27  Serial.println("time,heart_rate,breath_rate"); // Print CSV header
28  }
29
30  void loop() {
31      // Get radar data
32      radar.Breath_Heart();
33
34      // Check if there is a valid sensor report
35      if (radar.sensor_report != 0x00) {
36          // Get current timestamp
37          unsigned long currentMillis = millis();
38
39          // Initialize heart rate and breath rate with default values
40          int heartRate = -1; // Default value if no heart rate data
41          int breathRate = -1; // Default value if no breath rate data
42
43          // Process sensor report
44          switch (radar.sensor_report) {
45              case HEARTRATEVAL:
46                  heartRate = radar.heart_rate; // Update heart rate
47                  break;
48
49              case BREATHVAL:
50                  breathRate = radar.breath_rate; // Update breath rate
51                  break;
52          }

```

```

53
54 // Print the data as a CSV row
55 Serial.print(currentMillis);
56 Serial.print(",");
57 Serial.print(heartRate != -1 ? String(heartRate) : "-");
58 Serial.print(",");
59 Serial.println(breathRate != -1 ? String(breathRate) : "-");
60 }
61
62 delay(200); // Add time delay to avoid program jam
63 }

```

Isječak kôda 6..1: Primjer programskog koda konfiguracije osjetila za snimanje podataka u .csv formatu

Ovaj programski kod omogućuje povezivanje i komunikaciju između mikrokontrolera i samog osjetila za vitalne znakove. Na početku programa uspostavlja se serijska komunikacija sa osjetilom. Kasnije se uključuje ranije spomenuto odbrojavanje od 10 sekundi koje omogućuje pripremu scene za snimanje aktivnosti. Taj dio uključen je samo u slučaju ovog snimanja podataka te ga u stvarnom, implementiranom sustavu neće biti. U glavnoj petlji koja se izvršava moguće je vidjeti da se prikupljena mjerenja zapisuju u .csv formatu. Mjerenja koja nisu validna označavaju se sa "-". Na kraju programskog isječka postoji zadano kašnjenje u izvođenju programa od 200 milisekundi koje je preporuka proizvođača osjetila kako bi se osigurao ispravna komunikacija i prijenos mjerenja s osjetila na mikrokontroler.

6.3.2. Primjer podataka prije i poslije čišćenja

Sirovi podaci (.txt)

```

Starting in 10 seconds...
Starting in 10 seconds...
Starting in 9 seconds...
Starting in 8 seconds...

```

```
Starting in 7 seconds...
Starting in 6 seconds...
Starting in 5 seconds...
Starting in 4 seconds...
Starting in 3 seconds...
Starting in 2 seconds...
Starting in 1 seconds...
time,heart_rate,breath_rate
53 59 81 05 00 05 6F 83 AA C0 AD 40 54 43
10222,-,-
53 59 85 05 00 05 B0 B5 43
10422,-,-
53 59 81 02 00 01 0F 3F 54 43
10622,-,15
53 59 81 05 00 05 79 47 3B 5A 80 0C 54 43
11222,-,-
53 59 85 05 00 05 92 89 72 6F 7A B1 54 43
11423,-,-
53 59 81 05 00 05 8A 81 88 9F A1 0A 54 43
12224,-,-
53 59 85 05 00 05 76 7E 6D 8E 5F 89 54 43
12425,-,-
53 59 85 02 00 01 46 7A 54 43
12626,70,-
```

Očišćeni podaci (.csv)

```
time,heart_rate,breath_rate
10222,-,-
10422,-,-
10622,-,15
11222,-,-
11423,-,-
```


12224, -, -

12425, -, -

12626, 70, -

Kao što se može uočiti iz primjera podataka, čišćenjem sirovih mjerenja u heksadekadskom obliku i micanje početnog odbrojavanja prije početka snimanja mjerenja s osjetila dobiva se .csv format podataka koji se kasnije lakše obrađuje i analizira. Primjeri snimanja te sirovih i obrađenih mjerenja nalaze se u dodatku uz ovaj rad.

Ovim postupkom osigurana je ponovljivost eksperimenata i olakšana daljnja analiza podataka. Ova metoda omogućava da se snimljeni podaci lako koriste u simulaciji sustava razvijenoj u sljedećem poglavlju.

6.4. Osjetilo za prisutnost i pokrete

6.4.1. Opis osjetila i scenariji testiranja

24GHz mmWave osjetilo montirano je na strop prostorije dimenzija cca. 15 m² (5 × 3 m), na visini od 2,3 m. Fotografiju postavljenog osjetila montiranog na strop moguće je vidjeti na slici 6.2. Testirani scenariji uključivali su mikro-aktivnosti te aktivnosti na razini prostorije iz poglavlja 3. Kako se neke od aktivnosti preklapaju u načinu izvođenja (npr. sjedenje i čitanje te sjedenje i pletenje), takve aktivnosti su za potrebe ovih ispitivanja promatrane kao iste.



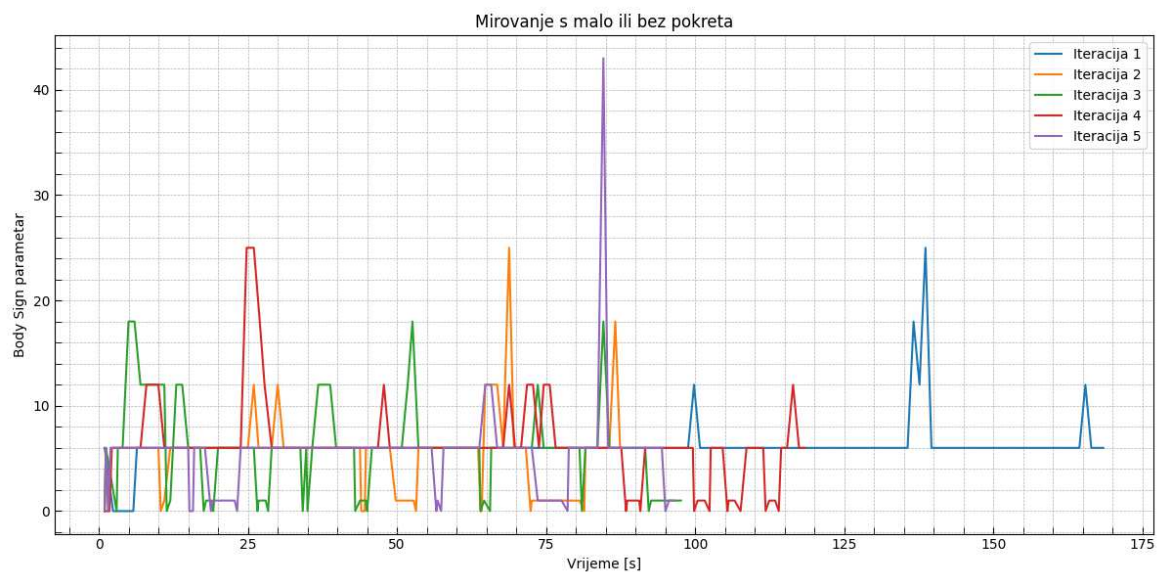
Slika 6.2. Fotografija postavljenog osjetila na stropu prostorije

Svaki scenarij sniman je u trajanju od otprilike 90 sekundi, a prikupljeni podaci analizirani su tako da su se prilikom izvođenja određenih aktivnosti gledale vrijednosti bodysign parametra koji može poprimiti vrijednosti od 1 do 100. Vrijednost 1 indikator je da je u prostoriji osoba koja se ne miče, dok su vrijednosti od 2 do 100 rezervirane za opisivanje pokreta osobe prisutne u prostoriji.

6.4.2. Zapažanja

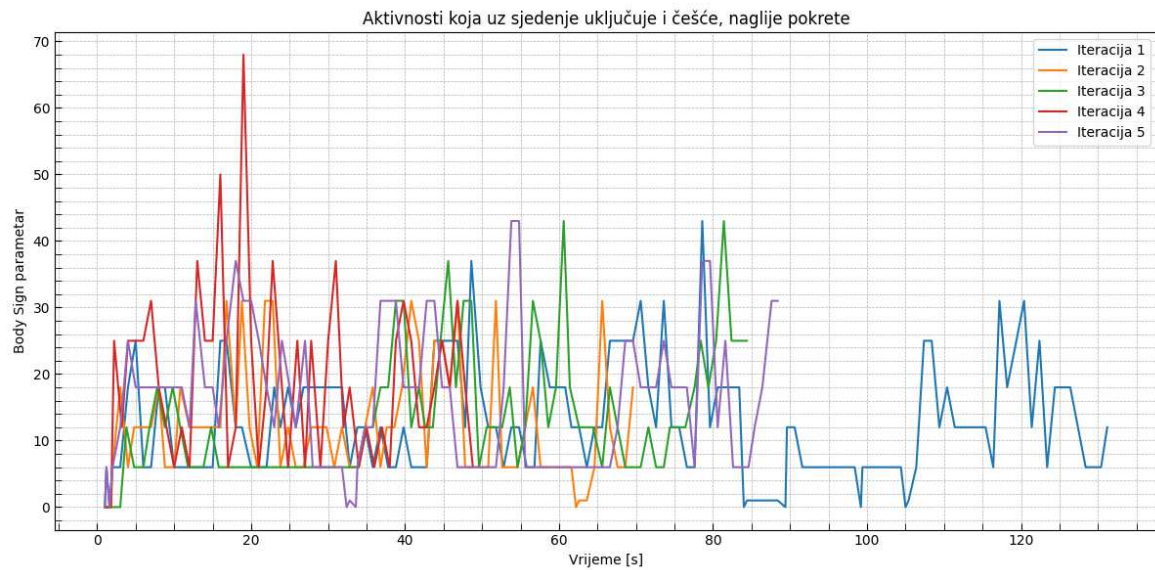
Od zapažanja bitno je izdvojiti vrijeme potrebno da osjetilo registrira izlazak iz prostorije koji se prilikom mjerenja kretao od 30 sekundi na više. Razlog tome je i sama tvornička konfiguracija osjetila u kojoj je određeno vrijeme nakon koje neaktivnost u prostoriji postaje stanje neprisutnosti u istoj. Pri višestrukim mjerenjima znalo je doći do lažnih mjerenja, osjetilo je registriralo stanje neprisutnosti prilikom kraćeg intervala mirovanja osobe, ali je nakon prvog manjeg pokreta vratio u stanje prisutnosti. To nije utjecalo na pouzdanost ove funkcionalnosti jer su vremenski razmaci od stanja neprisutnosti do ponovnog stanja prisutnosti bili veoma kratki (par sekundi). Također, zanimljivo je i to da osjetilo prepoz-

naje kada se osoba približava istom odnosno kada se udaljuje od njega. Što se tiče samog promatranja vrijednosti bodysign parametra kroz snimanja mjerenja uočene su ustaljene vrijednosti prilikom određenih aktivnosti. Na primjer, aktivnost "sjedenje i čitanje" koja se može poistovjetiti s aktivnošću "sjedenje i gledanje TV-a" te sličnima koje uključuju većinom statičan položaj osobe uz manje, ne toliko energične pokrete. Vrijednosti parametra bodysign_value prilikom mirovanja tj. sjedenja kreću se oko 6, dok se pri blažem pokretu osobe (listanje stranica, uzimanje daljinskog upravljača i slično) vrijednost parametra iznosi 12 uz poneke više vrijednosti (ako osoba brže posegne za daljinskim upravljačem ili se ustane po njega i slično). Vizualizaciju ovakvog tipa aktivnosti moguće je vidjeti na grafu 6.3. Na vizualizaciji je također moguće uočiti i trenutke kada osoba potpuno miruje te se vrijednost Body Sign parametra u tim trenucima očitava kao 0.



Slika 6.3. Vizualizacija mirne aktivnosti s malo ili bez pokreta

U usporedbi s prethodno prikazanom, mirnijom aktivnosti, aktivnost poput jedenja ili pak pletenja koje uključuju više pokreta pri izvođenju, vrijednost parametra bodysign iznosi u prosjeku od 18 do 30, ali moguće su i vrijednosti veće od 30 ako se osoba energično kretala u smjeru osjetila ili se pak pokretom približila osjetilu prilikom izvođenja tih naglijih pokreta. Vizualizaciju takve aktivnosti moguće je vidjeti na grafu 6.4.



Slika 6.4. Vizualizacija aktivnosti s naglijim pokretima (sjedenje i objed)

6.5. Osjetilo za praćenje vitalnih znakova

6.5.1. Opis osjetila i scenariji testiranja

60GHz mmWave osjetilo za praćenje vitalnih znakova testirano je kako bi se ispitala preciznost i konzistentnost u prepoznavanju otkucaja srca (`heart_rate`) i ritma disanja (`breath_rate`). Osjetilo je postavljeno u tri različite konfiguracije:

- **Na stropu:** Osjetilo je montirano na strop iznad kreveta na visini od 1,80 m od kreveta, kako bi se simulirala situacija u kojoj osjetilo pokriva cijeli prostor kreveta bez prepreka. Fotografiju postavljenog osjetila u ovoj konfiguraciji moguće je vidjeti na slici 6.2.
- **Iznad zaglavlja:** Osjetilo je montirano pod kutom te manje od 2 m na zidu iznad zaglavlja kreveta, što simulira scenarij u kojem se osjetilo koristi za praćenje vitalnih znakova dok osoba leži u krevetu. Prikaze ove konfiguracije moguće je vidjeti na slikama 6.5. i 6.6.
- **Horizontalno:** Osjetilo je postavljeno ispred osobe, primjerice na zidu ili iza ogledala, kako bi pokrивao dio tijela u sjedećem ili stojećem položaju. U testnoj okolini, osjetilo je postavljeno na udaljenost od cca. 50 cm od torza osobe kojoj se mjere vitalni znakovi.



Slika 6.5. Prikaz postavljenog osjetila iznad zaglavlja kreveta



Slika 6.6. Prikaz postavljenog osjetila iznad zaglavlja kreveta, uvećano kako bi se lakše uočilo osjetilo

Snimanje mjerenja je uključivalo dva scenarija za svaku od maloprije spomenutih konfiguracija pa su se tako promatrali slučajevi u kojima je osoba pokrivena dekom i slučajevi kada osoba nije pokrivena dekom kako bi se uočila moguća odstupanja u mjerenju prilikom nailaska na prepreku između osjetila i osobe. Također, mjerenja su se promatrala i kroz položaje u kojima se osoba nalazila pa su se tako razlikovale situacije u kojima osoba leži na boku te kada leži na leđima. Snimanje svake iteracije aktivnosti provedeno je u intervalu od otprilike 90 sekundi te su podaci nakon snimanja očišćeni te analizirani kroz vizualizacije istih. Mjerenja su također pokazala da se na osjetilu povremeno javljaju kratki intervali bez podataka (označeni s -), što može biti posljedica kašnjenja u obradi podataka. Ove vrijednosti izuzete su iz vizualizacija koje slijede u nastavku, a prilikom simulacije sustava opisanog u sljedećem poglavlju su zamijenjene posljednjim validnim mjerenjem.

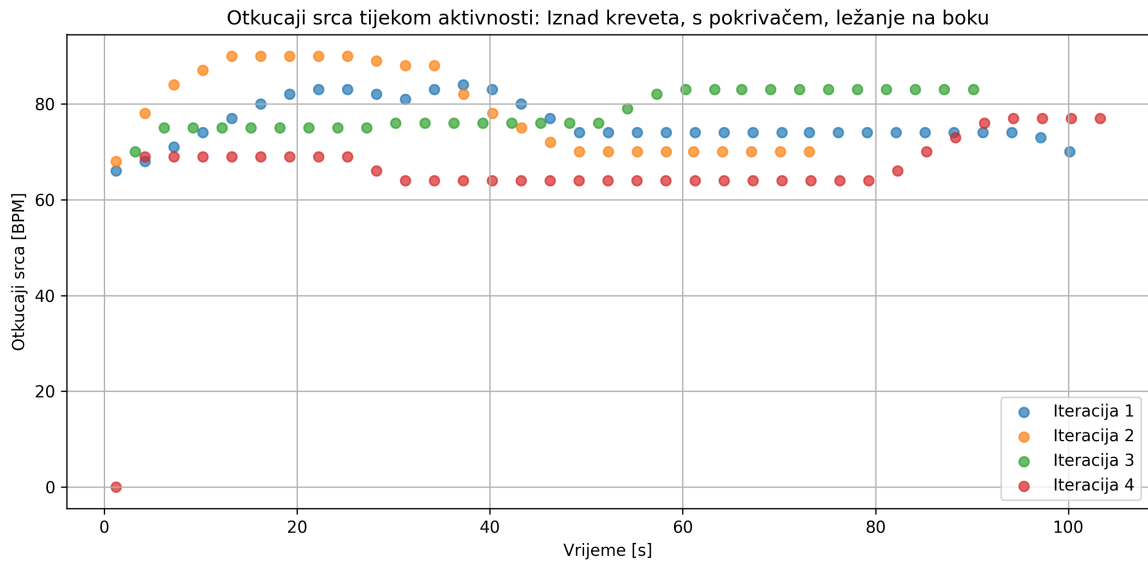
6.5.2. Zapažanja

Analiza vizualizacija podataka pruža jasnu razliku u performansama osjetila ovisno o lokaciji postavljanja osjetila. U nastavku su detaljna zapažanja za mjerenje otkucaja srca i ritma disanja na testiranim lokacijama osjetila.

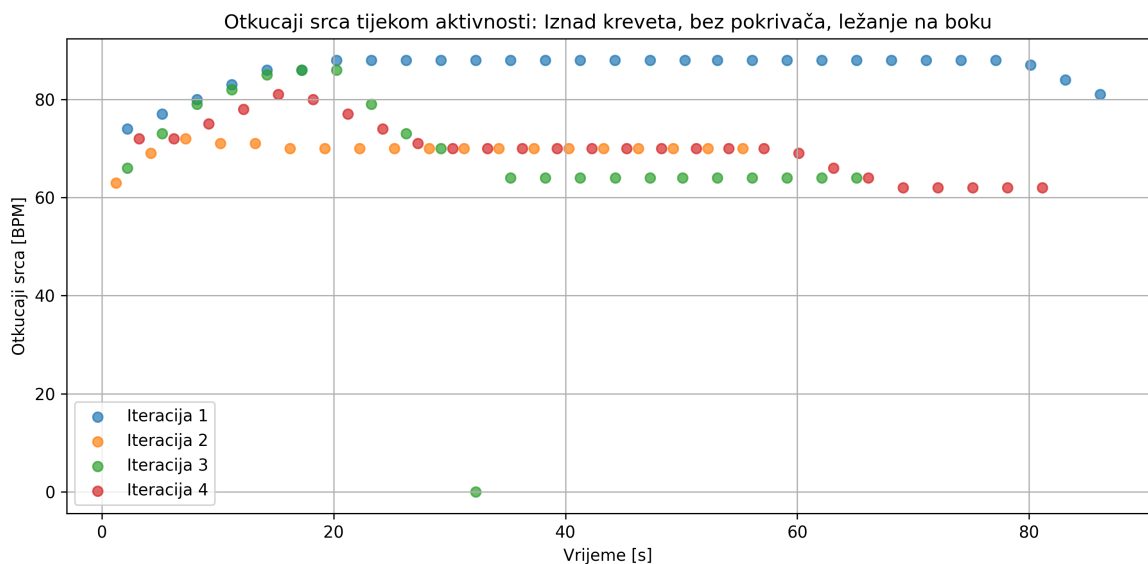
Otkucaji srca

Osjetilo montirano na stropu pokazao je nisku pouzdanost u mjerenju otkucaja srca. Podaci su prekidani šumom i sadrže nepravilnosti koje otežavaju praćenje trenda. Iako su prisutni neki segmenti s očitavanjima, veći dio snimanja pokazuje oscilacije koje ne odgovaraju očekivanim otkucajima srca.

Nasuprot tome, mjerenja s osjetila postavljenog iznad zaglavlja kreveta pokazala su najbolje rezultate. Ova lokacija osjetila omogućila je pouzdana mjerenja otkucaja, što ju čini najboljim izborom za primjenu u daljnjem razvoju sustava. Razlike u pouzdanosti mjerenja, prikazane na grafovima 6.7. i 6.8., nisu primijećene između situacija kada je osoba pokrivena dekom i kada nije, odnosno prisutnost prepreke između osobe i osjetila nije utjecala na pouzdanost mjerenja.



Slika 6.7. Vizualizacija mjerenja otkucaja srca s preprekom između osjetila i osobe

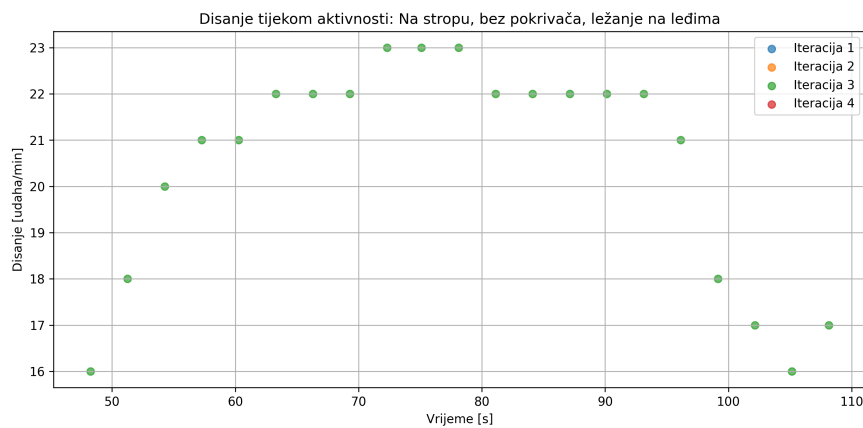


Slika 6.8. Vizualizacija mjerenja otkucaja srca bez prepreke između osjetila i osobe

Horizontalno montirano osjetilo pružio je također dobre rezultate. Podaci su uglavnom konzistentni, ali s povremenim segmentima šuma. Očitanja osjetila s ove lokacije bila su pouzdana te se ova lokacija također može koristiti u daljnjem razvoju ovog sustava.

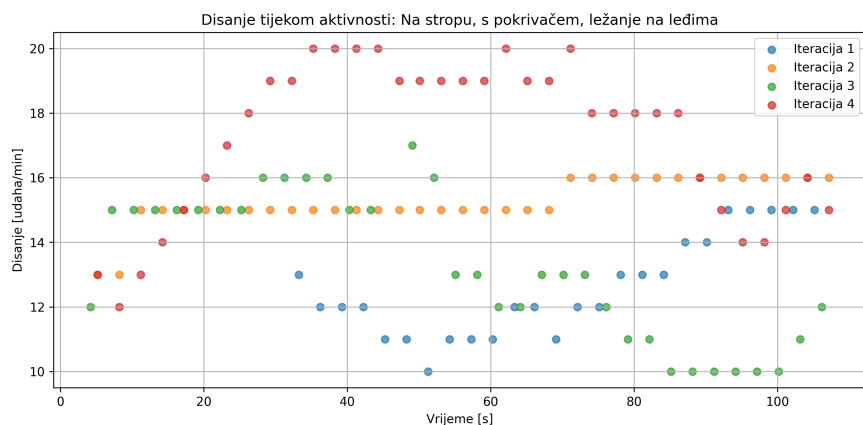
Ritam disanja

Mjerenja ritma disanja sa osjetila montiranog na stropu bila su nepouzdana. Većina mjerenja nije uspjela jasno zabilježiti oscilacije ritma disanja, što je moguće vidjeti na grafu 6.9. , gdje se također može uočiti da se na odabrane 4 iteracije snimanja očitavanja podataka u čak 3 od 4 iteracije ritam disanja nije ni ustanovio.



Slika 6.9. Vizualizacija mjerenja ritma disanja bez prepreke između osjetila i osobe, osjetilo montirano na stropu

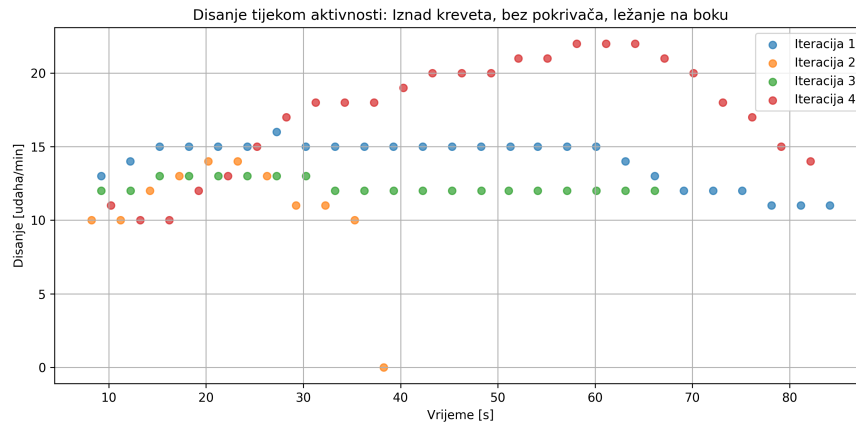
Zanimljivo, mjerenja su bila pouzdanija dok je osoba bila pokrivena dekom, što je moguće vidjeti na sljedećem grafu 6.10.



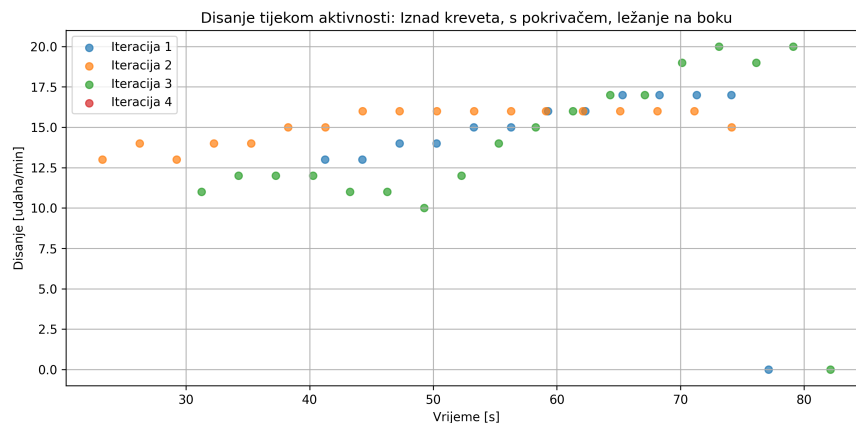
Slika 6.10. Vizualizacija mjerenja ritma disanja bez prepreke između osjetila i osobe, osjetilo montirano na stropu

Osjetilo montirano iznad zaglavlja kreveta ponovno se pokazao kao najbolja opcija. Mjerenja ritma disanja bila su uglavnom pouzdana, s jasnim i pravilnim oscilacijama koje odražavaju prirodni ritam disanja. Pouzdanost osjetila na ovoj

lokaciji bila je uglavnom zadovoljavajuća iako se na grafovima 6.11. i 6.12. može uočiti da za neke od iteracija nedostaju poneka, a u jednom slučaju čak i sva mjerenja tijekom jedne iteracije. Prepreka između osobe i osjetila nije znatno utjecala na pouzdanost mjerenja ritma disanja.

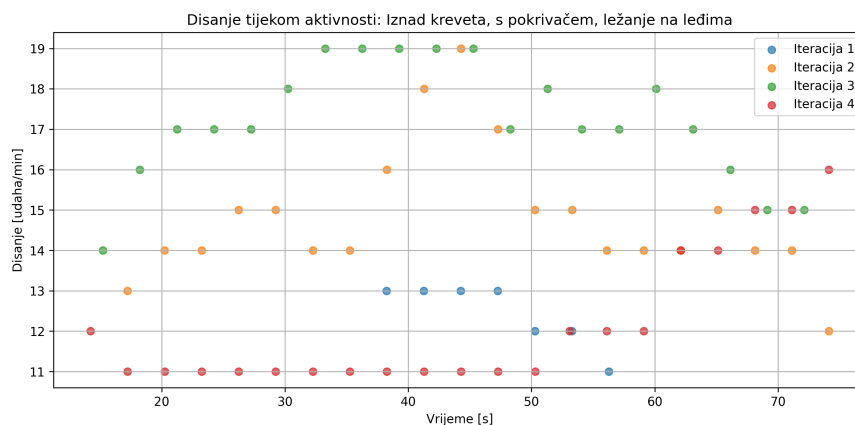


Slika 6.11. Vizualizacija mjerenja ritma disanja bez prepreke između osjetila i osobe, osjetilo montirano iznad zaglavlja kreveta



Slika 6.12. Vizualizacija mjerenja ritma disanja s preprekom između osjetila i osobe, osjetilo montirano iznad zaglavlja kreveta

Položaj u kojem se nalazila osoba za vrijeme mjerenja ritma disanja utjecao je na pouzdanost mjerenja pa se tako smanjuje pouzdanost mjerenja ukoliko osoba leži na boku, a ne na leđima kako je prikazano u grafovima 6.11. i 6.12. Mjerenja ritma disanja dok osoba leži na leđima moguće je vidjeti na grafu 6.13. Razlike u pouzdanosti mjerenja sa i bez prepreke (pokrivača) između osobe i osjetila su zanemarive.



Slika 6.13. Vizualizacija mjerenja ritma disanja s preprekom između osjetila i osobe, osjetilo montirano iznad zaglavlja kreveta

Horizontalno montirano osjetilo pokazao je zadovoljavajuću pouzdanost u mjerenju ritma disanja. Mjerenja su bila konzistentna kada je osoba mirovala, ali i kada se osoba blago gibala ispred osjetila.

6.6. Osjetilo za detekciju pada

6.6.1. Opis osjetila i scenariji testiranja

60GHz mmWave osjetilo montirano je na strop prostorije, na visini od 2,3 m. Scenariji testiranja uključivali su simulaciju pada kroz promjene položaja tijela iz stojećeg u ležeći položaj.

6.6.2. Zapažanja

Unatoč višestrukim pokušajima simuliranja pada, osjetilo nije uspješno detektirao promjene. Ovo ograničenje ukazuje na potrebu za dodatnom kalibracijom i optimizacijom.

6.7. Rezultati ispitivanja osjetila

Osjetilo za prisutnost i pokrete pokazalo se pouzdanim u detekciji prisutnosti i osnovnih aktivnosti u prostoriji, uz minimalne lažne pozitivne ili negativne detekcije. Problemi s detekcijom neprisutnosti rezultat su tvorničkih postavki koje je moguće prilagoditi ovisno o potrebama primjene. Parametar (`bodysign_val`)

omogućio je razlikovanje aktivnosti s većom i manjom razinom pokreta. Vrijednosti su se kretale unutar očekivanih granica (1–100), s dobrim odgovaranjem na promjene u intenzitetu pokreta unutar promatrane aktivnosti. Vrijeme reakcije osjetila na ulazak je bilo vrlo nisko i zadovoljavajuće dok je vrijeme reakcije osjetila na izlazak iz prostorije, iako ponekad produženo, nije značajno utjecalo na pouzdanost prikupljenih podataka, no moglo bi biti optimizirano u daljnjem razvoju sustava. Osjetilo se pokazalo posebno korisnim za praćenje aktivnosti u prostoriji gdje su dominantni statični položaji (sjedenje, ležanje), uz mogućnost razlikovanja osnovnih tipova aktivnosti na temelju intenziteta pokreta.

Osjetilo za praćenje vitalnih znakova, otkucaja srca te ritma disanja, najpouzdanija mjerenja očitavalo je montirano iznad zaglavlja kreveta. Lokacija montiranja osjetila na strop prostorije isprobana je zbog mogućnosti osjetila da detektira i pokrete. U tom slučaju, jedno osjetilo moglo bi se koristiti za praćenje i vitalnih znakova i prisutnosti u prostoriji, prebacujući se između načina rada za vitalne znakove i prisutnost. Ipak, mjerenja sa stropa pokazala su lošu pouzdanost u usporedbi s drugim konfiguracijama te je ta zamisao u odbačena kroz daljnji razvitak ovog sustava. Prepreke između osobe i osjetila poput pokrivača nisu značajno utjecale na preciznost mjerenja vitalnih znakova, iako su povremeno bilježene manje oscilacije u mjerenjima. U testiranjima su također provedena mjerenja u dva različita scenarija odnosno u dvije pozicije u kojima se osoba može naći dok leži u krevetu: dok je osoba ležala na boku i na leđima, pri čemu nije uočena značajna razlika koja bi utjecala na kvalitetu mjerenja s obzirom na poziciju tijela osobe. Ovaj zaključak sugerira da je osjetilo donekle prilagodljivo različitim položajima tijela prilikom praćenja vitalnih znakova. Važno je napomenuti da mjerenja vitalnih znakova mogu odstupati od stvarnih vrijednosti. Nije bilo mogućnosti usporedbe stvarnih vrijednosti otkucaja srca i ritma disanja s onima koje je osjetilo očitavalo. Ovo ograničenje otežava točnu procjenu preciznosti osjetila, a buduća testiranja trebala bi uključivati reference izravnog mjerenja vitalnih znakova.

Osjetilo za detekciju pada, nažalost, nije uspješno ispitano te trenutno nije moguće nastaviti daljnji razvoj sustava s ovim osjetilom.

7. Opis razvijenog sustava

U ovom poglavlju detaljno će biti opisan sustav za praćenje aktivnosti osoba u kućanstvu. Biti će opisana mreža osjetila, način prikupljanja i obrade podataka s osjetila, konfiguracija kućanstva, arhitektura centralnog računala i njegove funkcionalnosti, frontend sučelje za vizualizaciju podataka te podrška za stvarna osjetila. Također će biti objašnjen simulator aktivnosti koji omogućuje testiranje sustava bez fizičkih osjetila.

7.1. Pregled arhitekture sustava

Sustav za praćenje aktivnosti osoba u kućanstvu sastoji se od sljedećih komponenti:

- **Centralno računalo** – Poslužitelj implementiran u programskom jeziku Python koristeći FastAPI radni okvir. Ovaj poslužitelj obrađuje podatke s osjetila, upravlja stanjem prostorija i korisnika te omogućuje komunikaciju s osjetilima putem HTTP protokola te s klijentima putem HTTP i WebSocket protokola.
- **Osjetila** – Prava osjetila povezana na ESP32 mikroupravljač i simulirana osjetila generirana unutar testnog okruženja. Osjetila mogu detektirati prisutnost i pokrete te mjeriti vitalne znakove.
- **Simulator aktivnosti** – Programski alat koji čita unaprijed snimljena mjerenja s osjetila pohranjena unutar CSV datoteka te ih šalje na poslužitelj kako bi se simulirao rad unutar testnog okruženja fizičkih osjetila bez potrebe da se koriste stvarna osjetila.

- **Frontend sučelje** – Web sučelje koje vizualizira podatke u stvarnom vremenu. Povezano je s centralnim računalom sustavom putem WebSocket protokola.

7.2. Tok podataka u sustavu

Podaci se kreću kroz sustav na sljedeći način:

1. **Osjetila** (ESP32 ili simulator) periodično šalju podatke u JSON formatu putem HTTP POST zahtjeva prema centralnom računalu.
2. **Centralno računalo** obrađuje podatke i ažurira stanje prostorija i korisnika.
3. **Ažurirani podaci** distribuiraju se prema frontend sučelju u stvarnom vremenu putem WebSocket protokola.

7.3. Mreža osjetila

Mreža osjetila sastoji se od dva tipa osjetila:

- **Osjetila prisutnosti i pokreta (HSP)** – detektiraju prisutnost i kretanje osoba unutar prostorija.
- **Osjetila vitalnih znakova (HA)** – mjere otkucaje srca i ritam disanja osobe.

Sama mreža osjetila može sadržavati i druga osjetila, što zahtjeva minimalne izmjene u kasnije opisanom poslužitelju.

7.4. Konfiguracija kućanstva

```
1
2 {
3   "sensors": [
4     {
5       "id": "HSP_001",
6       "type": "presence_and_motion",
```

```

7     "room": "Living Room",
8     "columns": ["time", "status", "bodysign_val"]
9 },
10 {
11     "id": "HA_001",
12     "type": "heart_and_breath",
13     "room": "Living Room",
14     "columns": ["time", "heart_rate", "breath_rate"]
15 }
16 ]
17 }

```

Isječak kôda 7.1: Primjer konfiguracije kućanstva definirane u household_config.json

Konfiguracija kućanstva definirana je u JSON datoteci household_config.json kao u gore prikazanom primjeru, gdje se navode sva osjetila unutar kućanstva. Svako osjetilo ima jedinstveni identifikator id, pridruženu prostoriju i listu parametara koje mjeri. U prostorijama može postojati više osjetila različitih tipova, ali samo jedno osjetilo istog tipa. U slučaju da u kućanstvu postoji toliko velika soba da jedno osjetilo ne može obuhvatiti istu sa svojim dometom detekcije, takva prostorija dijeli se u sekcije. Svaka sekcija u tom slučaju predstavlja "zasebnu" prostoriju. Primjer takve konfiguracije moguće je vidjeti u nastavku.

```

1
2 {
3     "sensors": [
4         {
5             "id": "HSP_001",
6             "type": "presence_and_motion",
7             "room": "Living_Room_1",
8             "columns": ["time", "status", "bodysign_val"]
9         },
10        {

```

```

11     "id": "HSP_002",
12     "type": "presence_and_motion",
13     "room": "Living_Room_2",
14     "columns": ["time", "status", "bodysign_val"]
15 }
16 ]
17 }

```

Isječak kôda 7..2: Primjer konfiguracije veće prostorije podijeljene na više dijelova definirane u `household_config.json`

Ovim pristupom omogućava se korištenje osjetila s ograničenim dometom u velikim prostorima bez narušavanja logike sustava.

7.5. Centralno računalo

Centralno računalo upravlja mjerenjima pristiglim s osjetila, određuje stanje prostorija te prati psihofizičko stanje korisnika. Implementirano je u programskom jeziku Python koristeći **FastAPI** radni okvir i omogućuje obradu mjerenja s osjetila te dvosmjernu komunikaciju s web sučeljem putem WebSocket protokola.

Upravljanje podacima s osjetila

Osjetila periodično šalju mjerenja poslužitelju u JSON formatu, a poslužitelj ih obrađuje i ažurira trenutačno stanje prostorija. Svako osjetilo ima pridruženu određenu prostoriju i njegova mjerenja utječu na promjenu stanja te prostorije. Primanje podataka implementirano je s pomoću HTTP POST krajnje točke:

```

1
2 @app.post("/send_data")
3 async def receive_sensor_data(request: Request):
4     # Receive sensor data from the request
5     sensor_data = await request.json()
6     sensor_id = sensor_data.get("sensor_id")
7
8     # Check if sensor_id is provided in the data

```

```

9     if not sensor_id:
10         raise HTTPException(status_code=400, detail="sensor_id is
           ↪ required")
11
12     # Handle the received sensor data
13     handle_sensor_data(sensor_id, sensor_data)
14
15     # Broadcast the update to all connected clients
16     await broadcast_update(sensor_id, sensor_data)
17
18     # Return a success response
19     return {"status": "success", "sensor_id": sensor_id}

```

Isječak kôda 7..3: Definicija krajnje točke za slanje mjerenja s osjetila na poslužitelj

Prilikom primitka podataka, poziva se funkcija `handle_sensor_data`, koja na temelju vrste osjetila ažurira stanje prostorije ili psihofizičko stanje korisnika.

```

1
2 def handle_sensor_data(sensor_id, data):
3     # Retrieve the sensor object using the sensor_id
4     sensor = room_manager.sensors.get(sensor_id)
5
6     # If the sensor is not found, print a warning and return
7     if not sensor:
8         print(f"Warning: Unknown sensor ID {sensor_id}")
9         return
10
11     # Filter out any data entries with a value of '-'
12     data = {
13         key: value for key, value in data.items() if value != '-'
14     }
15
16     # Handle the sensor data based on the sensor type
17     if sensor.type == SensorType.HSP.value:

```



```

18     handle_hsp_sensor(sensor_id, data)
19     elif sensor.type == SensorType.HA.value:
20         handle_ha_sensor(sensor_id, data)
21     else:
22         print(f"Unsupported sensor type: {sensor.type}")
23
24     # Update the state of the person based on the new sensor data
25     person_manager.update_state(room_manager)
26
27     # Print the updated physiological state and last known room of the
28     ↪ person
    print(f"Updated Person State: {person_manager.physiological_state},
    ↪ Room: {person_manager.last_known_room}")

```

Isječak kôda 7..4: Funkcija `handle_sensor_data`

Praćenje stanja zauzetosti prostorija

Svaka prostorija može biti u jednom od dva stanja: *zauzeta* (OCCUPIED) ili *prazna* (EMPTY). Poslužitelj određuje stanje prostorije na temelju podataka osjetila prisutnosti i osjetila vitalnih znakova.

Osjetilo prisutnosti i pokreta (HSP) ima glavnu ulogu u određivanju zauzetosti prostorije. Njegova obrada funkcionira na sljedeći način:

```

1
2 def handle_hsp_sensor(sensor_id, data):
3     # Retrieve the sensor object using the sensor_id
4     sensor = room_manager.sensors[sensor_id]
5
6     # Extract status and bodysign_val from the data
7     status = data.get("status")
8     bodysign_val = data.get("bodysign_val")
9
10    # Determine the room state based on the status and bodysign_val
11    room_state = RoomState.EMPTY if status == "NOONE" or (

```

```

12     status in ["NOTHING", "BODYVAL"] and bodysign_val == 0) else
13         ↪ RoomState.OCCUPIED
14
15     # Update the room state in the room manager
16     room_manager.update_room_state(sensor.room, room_state)
17
18     # Add the sensor data to the room manager
19     room_manager.add_sensor_data(sensor_id, data)
20
21     # If the room state is occupied, clear occupancy from all HA-only
22     ↪ rooms
23
24     if room_state == RoomState.OCCUPIED:
25         for other_room in room_manager.rooms.values():
26             if other_room.room != sensor.room:
27                 # Check if there is any HSP sensor present in the other
28                 ↪ room
29                 hsp_present = any(
30                     s for s in room_manager.sensors.values()
31                     if s.room == other_room.room and s.type ==
32                     ↪ SensorType.HSP.value
33                 )
34
35                 # If the room has only HA sensors (no HSP), mark it as
36                 ↪ empty
37
38                 if not hsp_present:
39                     room_manager.update_room_state(
40                         other_room.room, RoomState.EMPTY)

```

Isječak kôda 7..5: Metoda za određivanje zauzetosti prostorije na temelju podataka s osjetila prisutnosti i pokreta

Ova funkcija analizira status osjetila prisutnosti i donosi odluku je li prostorija zauzeta ili ne. Ako osjetilo detektira prisutnost tijela, prostorija se označava kao zauzeta (OCCUPIED). Ako podaci osjetila pokazuju da nema nikoga (NOONE) ili ako su izmjerene vrijednosti bodysign parametra 0 (bodysign_val = 0), tada se

prostorija označava kao prazna (EMPTY).

Kako bi se poboljšala funkcionalnost otkrivanja zauzete prostorije odnosno određivanja prostorije u kojoj se osoba nalazi, implementirana je dodatna provera podataka s osjetila otkucaja srca i ritma disanja. Ovo osjetilo može pomoći u identificiranju osoba u prostoriji kada nema podataka s osjetila prisutnosti.

```
1
2 def handle_ha_sensor(sensor_id, data):
3     # Get the room associated with the sensor
4     room = room_manager.sensors[sensor_id].room
5
6     # Add the sensor data to the room
7     room_manager.add_sensor_data(sensor_id, data)
8
9     # Extract heart rate and breath rate from the data
10    heart_rate = int(data.get("heart_rate", 0))
11    breath_rate = int(data.get("breath_rate", 0))
12
13    # Check if any HSP sensor is active in another room
14    hsp_active_rooms = [
15        room.room for room in room_manager.rooms.values()
16        if any(
17            sensor for sensor in room.sensor_reading.values()
18            if sensor.sensor.type == SensorType.HSP.value and room.state
19            ↪ == RoomState.OCCUPIED
20        )
21    ]
22
23    # Ignore HA sensor updates if any HSP sensor is active
24    if hsp_active_rooms:
25        return
26
27    # Only mark occupancy if no HSP sensors detected presence and vital
28    ↪ signs are present
```

```
27     if heart_rate > 0 and breath_rate > 0:
28         room_manager.update_room_state(room, RoomState.OCCUPIED)
```

Isječak kôda 7..6: Metoda kojom se s pomoću osjetila za vitalne znakove određuje zauzetost prostorije

Ova funkcija obrađuje podatke s osjetila otkucaja srca i ritma disanja i provjerava postoji li već osjetilo prisutnosti i pokreta u istoj prostoriji. Ako u prostoriji postoji osjetilo prisutnosti, ono ima prednost pri određivanju stanja prostorije, a podaci s osjetila otkucaja srca i ritma disanja neće utjecati na status zauzetosti prostorije.

Međutim, ako osjetila prisutnosti nema, tada se stanje prostorije određuje na temelju izmjerenih vrijednosti otkucaja srca i ritma disanja. Ako postoje podaci o vitalnim znakovima osobe, prostorija se označava kao zauzeta odnosno OCCUPIED. U suprotnom, prostorija ostaje prazna to jest EMPTY.

Ova nadogradnja poboljšava otkrivanje prisutnosti u prostorijama gdje osjetilo pokreta možda ne daje dovoljno informacija ili ga nema u prostoriji, omogućujući pouzdanije praćenje stanja prostorija.

Određivanje trenutnog psihofizičkog stanja korisnika

Sustav također prati fiziološke parametre korisnika, poput otkucaja srca i ritma disanja. Na temelju tih podataka poslužitelj može odrediti je li korisnik u normalnom stanju ili postoji razlog za zabrinutost:

```
1
2 class PersonManager:
3     physiological_state: str = "Not yet determined"
4     last_known_room: str = None
5
6     # Update the physiological state based on sensor readings
7     def update_physiological_state(self, sensor_reading: SensorReading):
8         heart_rate = sensor_reading.data.get("heart_rate")
9         breath_rate = sensor_reading.data.get("breath_rate")
10
```

```

11     # Check if heart rate data is available
12     if heart_rate is None:
13         self.physiological_state = "Not enough vital signs data -
14             ↪ heart rate"
15         return
16
17     # Check if breath rate data is available
18     if breath_rate is None:
19         self.physiological_state = "Not enough vital signs data -
20             ↪ breath rate"
21         return
22
23     # Determine physiological state based on heart rate and breath
24     ↪ rate
25     if int(heart_rate) > 90 and int(breath_rate) > 20:
26         self.physiological_state = "Suspicious"
27     elif int(heart_rate) <= 90 and int(breath_rate) <= 20:
28         self.physiological_state = "Normal"
29     else:
30         self.physiological_state = "Unknown"

```

Isječak kôda 7..7: Implementacija klase koja, na temelju vitalnih znakova osobe, određuje psihofizičko stanje

Ova klasa analizira podatke s osjetila vitalnih znakova i na temelju unaprijed definiranih pragova određuje stanje korisnika. Ako su vrijednosti izvan granica normalnog, stanje se označava kao sumnjivo to jest Suspicious, dok se normalne vrijednosti označavaju kao normalne. Važno je napomenuti da je ovo vrlo primitivna metoda za određivanje psihofizičkog stanja te se takva neće koristiti u stvarnom sustavu. Ovako je implementirana u svrhu testiranja samog rada sustava te se predlaže nadogradnja u budućem radu.

Dohvat stanja prostorija i korisnika

Poslužitelj omogućuje dohvat trenutnog stanja kućanstva i korisnika putem HTTP GET krajnje točke:

```
1
2 @app.get("/current_status")
3 async def get_current_status():
4     """
5     Endpoint to get the current status of the household and person.
6     Combines the household state and person state into a single response.
7     """
8     response = {
9         "household_state": await get_household_state(),
10        "person_state": await get_person_state(),
11    }
12    return response
```

Isječak kôda 7.8: HTTP krajna točka s pomoću koje se s poslužitelja dohvaća trenutno stanje kućanstva i osobe

Ova krajnja točka vraća trenutno stanje svih prostorija kao i psihofizičko stanje korisnika, omogućujući prikaz podataka u stvarnom vremenu unutar web sučelja.

7.6. Simulator aktivnosti

Kako bi se omogućilo testiranje sustava bez potrebe za stvarnim fizičkim osjetilima, razvijen je simulator aktivnosti koji generira i šalje podatke prema poslužitelju. Simulator koristi stvarne snimljene podatke pohranjene u CSV datotekama te simulira rad osjetila oponašajući njihovo prirodno ponašanje u različitim vremenskim intervalima.

Struktura simulatora aktivnosti

Simulator aktivnosti sastoji se od nekoliko ključnih komponenti:

- **Konfiguracijska datoteka kućanstva (household_config.json)** – opisuje osjetila unutar kućanstva i njihove karakteristike.
- **Konfiguracijska datoteka simulacije (simulator_config.json)** – definira scenarije i aktivnosti koje simulator reproducira.
- **Mehanizam simulacije osjetila (sensor_simulation.py)** - očitava unaprijed snimljena mjerenja s osjetila iz CSV datoteka i šalje ih poslužitelju uz prilagođene vremenske razmake.
- **Mehanizam simulacije aktivnosti (run_sensor.py)** – čita aktivnosti iz konfiguracijske datoteke simulacije te ih s pomoću mehanizma za simulaciju osjetila slijedno izvršava.

Svaka aktivnost definira koja se osjetila koriste i koje CSV datoteke sadrže podatke koji će se slati na poslužitelj. Primjer jednog scenarija iz konfiguracijske datoteke simulator_config.json gdje osoba ulazi u dnevni boravak te nakon ulaska u prostoriju sjedi i čita knjigu:

```

1 {
2   "activities": [
3     {
4       "activity": "entering_living_room",
5       "description": "Person enters the living room",
6       "sensors": [
7         {
8           "id": "HSP_001",
9           "file": ".../entering_the_room_and_sitting_2.csv"
10        }
11      ]
12    },
13    {
14      "activity": "sitting_in_living_room",
15      "description": "Person is sitting and reading",
16      "sensors": [

```

```

17     {
18         "id": "HSP_001",
19         "file": ".../sitting_and_reading.csv"
20     },
21     {
22         "id": "HA_001",
23         "file": ".../HA1_horizontal_mounted_4.csv"
24     }
25 ]
26 }
27 ]
28 }

```

Isječak kôda 7..9: Prikaz scenarija u kojem osoba sjedi i čita nakon što je ušla u dnevni boravak

Pokretanje simulacije

Mehanizam simulacije aktivnosti (`run_sensor.py`) učitava konfiguraciju osjetila unutar kućanstva iz datoteke na sljedeći način:

```

1
2 def run_simulation(config_file, simulator_config_file):
3     # Load the household configuration from the specified file
4     with open(config_file, "r") as f:
5         config = json.load(f)
6
7     # Load the simulator configuration from the specified file
8     with open(simulator_config_file, "r") as f:
9         simulator_config = json.load(f)
10
11     # Define the server endpoint for receiving sensor data
12     # Replace <SERVER_ADDRESS> with actual server address
13     endpoint = "http://<SERVER_ADDRESS>/send_data"
14

```



```

15     # Iterate through each activity in the simulator configuration
16     for activity in simulator_config["activities"]:
17         # Simulate the activity with the given configuration and endpoint
18         simulate_activity(activity, config, endpoint)

```

Isječak kôda 7..10: Metoda za pokretanje simulacije s pomoću unaprijed zadanog scenarija aktivnosti

Zatim mehanizam za simulaciju osjetila (`sensor_simulation.py`) simulira svoje očitavanje čitajući redak po redak iz CSV datoteke te šalje podatke na poslužitelj koristeći HTTP POST zahtjeve:

```

1 def simulate_sensor(sensor_id, sensor_type, csv_file, columns, endpoint):
2     """Simulate sensor data by reading from a CSV file."""
3     # Read the CSV file and select the specified columns
4     data = pd.read_csv(csv_file, usecols=columns)
5
6     # Iterate over each row in the data
7     for i, row in data.iterrows():
8         # Create a payload dictionary with the sensor data
9         payload = {col: row[col] for col in columns}
10        payload["sensor_id"] = sensor_id
11        payload["type"] = sensor_type
12
13        ...
14
15        # Send the payload to the specified endpoint
16        response = requests.post(endpoint, json=payload)
17        print(f"Sensor {sensor_id} sent: {payload}, Response:
18              ↪ {response.status_code}")
19
20        # Wait for the calculated time before sending the next data point
21        time.sleep(wait_time)

```

Isječak kôda 7..11: Metoda s pomoću koje se mjerenja s osjetila slijedno čitaju te šalju na poslužitelj

Paralelno pokretanje simulacije osjetila

Kako bi se omogućila istovremena simulacija više osjetila unutar jedne aktivnosti, za svako osjetilo pokreće se zaseban proces koristeći multiprocessing modul. To znači da, ukoliko u jednoj prostoriji postoji više osjetila, za očekivati je da će različita osjetila u istom trenutku poslati svoja mjerenja, a sustav mora biti spreman obraditi mjerenja s više osjetila u istom trenutku. Primjer takvog ponašanja osjetila moguće je uočiti na prikazu konfiguracije simulatora aktivnosti, gdje se unutar iste aktivnosti simuliraju paralelno mjerenja i sa osjetila prisutnosti i pokreta te osjetila vitalnih znakova. Na taj način, više osjetila može slati podatke paralelno, čime se na realističan način oponaša rad osjetila te njihovog slanja podataka na centralno računalo, što je moguće vidjeti u sljedećem isječku koda mehanizma za simulaciju aktivnosti (`run_sensor.py`):

```
1 # Simulate a single sensor based on its configuration.
2 def start_sensor_simulation(sensor_config, sensor_meta, endpoint):
3     sensor_id = sensor_config["id"]
4     sensor_type = sensor_meta["type"]
5     room = sensor_meta["room"]
6     file = sensor_config["file"]
7
8     # Print the details of the sensor and specific activity being
9     ↪ simulated
10    print(f"Simulating sensor {sensor_id} of type {sensor_type} in {room}
11    ↪ using file {file}")
12
13    # Call the simulate_sensor function with the appropriate parameters
14    simulate_sensor(sensor_id, sensor_type, file, sensor_meta["columns"],
15    ↪ endpoint)
16
17 # Simulate all sensors in a given activity.
18 def simulate_activity(activity, config, endpoint):
19     # Print the description of the activity being started
20     print(f"\nStarting activity: {activity['description']}")
```

```

19
20 # List to keep track of all processes
21 processes = []
22
23 # Iterate over each sensor in the activity and start if sensor is in
   ↳ the household configuration
24 for sensor in activity["sensors"]:
25     # Find metadata for the sensor in household_config.json
26     sensor_meta = None
27     for s in config["sensors"]:
28         if s["id"] == sensor["id"]:
29             sensor_meta = s
30             break
31
32     # If sensor metadata is not found, print a warning and skip this
   ↳ sensor
33     if not sensor_meta:
34         print(f"Warning: Sensor {sensor['id']} not found in
   ↳ config.json. Skipping.")
35         continue
36
37     # Start a process to simulate the sensor
38     p = Process(target=start_sensor_simulation, args=(sensor,
   ↳ sensor_meta, endpoint))
39     processes.append(p)
40     p.start()
41
42     # Wait for all sensor processes to complete
43     for p in processes:
44         p.join()

```

Isječak kôda 7.12: Metode za omogućavanje paralelnog izvođenja višestrukih osjetila unutar jedne aktivnosti

Ovaj način omogućuje istovremeno slanje podataka s više osjetila u istoj pros-

toriji, poboljšavajući oponašanje stvarnih situacija u sustavu.

Unutar mehanizma za simulaciju aktivnosti (`run_sensor.py`) odnosno oponašanje unaprijed određenog scenarija aktivnosti kroz konfiguracijsku datoteku simulatora aktivnosti opisanu iznad, pri samom pokretanju simulatora moguće je odabrati parametre poput adrese poslužitelja na koji će se mjerenja s osjetila odašiljati te je moguće odabrati željene konfiguracijske datoteka kućanstva i samog simulatora aktivnosti. To je moguće vidjeti u sljedećem isječku koda:

```
1
2 def run_simulation():
3     # Path to the household configuration file
4     config_file = "../household_configuration/household_config.json"
5
6     # Path to the simulator configuration file
7     simulator_config_file = "./simulator_config.json"
8
9     # Server endpoint for receiving sensor data, replace with link
10    ↳ provided by the server
11    endpoint = "http://<SERVER_ADDRESS>/send_data"
12
13    # Load household configuration from file
14    with open(config_file, "r") as f:
15        config = json.load(f)
16
17    # Load simulator configuration from file
18    with open(simulator_config_file, "r") as f:
19        simulator_config = json.load(f)
20
21    # Simulate each activity defined in the simulator configuration
22    for activity in simulator_config["activities"]:
```

Isječak kôda 7.13: Metode za omogućavanje paralelnog izvođenja višestrukih osjetila unutar jedne aktivnosti

Izračun vremenskog razmaka između očitavanja

Jedna od funkcionalnosti simulatora je dinamičko izračunavanje vremenskog razmaka između uzastopnih očitavanja s osjetila. Umjesto slanja podataka u fiksnim vremenskim intervalima, mehanizam za simulaciju osjetila (`sensor_simulation.py`) računa razmak između uzastopnih zapisa iz CSV datoteka i simulira stvarne vremenske tokove podataka:

```
1 # Calculate wait time (convert milliseconds to seconds)
2 if i < len(data) - 1:
3     # Get the time of the next row in the data
4     next_time = data.iloc[i + 1]["time"]
5     # Calculate the wait time by subtracting the current row's time from
6     # ↪ the next row's time
7     # and convert milliseconds to seconds. Ensure wait time is not
8     # ↪ negative by using max(0, ...)
9     wait_time = max(0, (next_time - row["time"]) / 1000.0)
10    # Uncomment the following line to print debug information about the
11    # ↪ times and wait time
12    # print(f"Next time: {next_time}, Current time: {row['time']}, Wait
13    # ↪ time: {wait_time}")
14 else:
15     # If this is the last row, set a default wait time of 0.1 seconds
16     wait_time = 0.1
```

Isječak kôda 7..14: Izračun vremenske razlike između dva mjerenja pristigla s osjetila

Ovaj pristup omogućava preciznu simulaciju rada osjetila, uz očuvanje prirodnih vremenskih odstupanja između očitavanja. Vremenski interval između dva mjerenja podijeljen je s 1000 jer su se u snimke mjerenja upisivale milisekunde umjesto sekundi radi preciznosti.

7.7. Podrška za stvarna osjetila

Sustav podržava stvarna osjetila povezana na ESP32 mikrokontroler koji komunicira putem WiFi mreže i periodično šalje podatke o prisutnosti i pokretima te vitalnim znakovima korisnika poslužitelju putem HTTP POST zahtjeva, ovisno o kojem je osjetilu riječ. Frekvencija slanja podataka nije unaprijed definirana, već osjetilo šalje podatke u trenutku kada očita nova mjerenja. Raspon vremenske razlike za očitavanja s osjetila za vitalne znakove je 1 - 30 sekundi, dok za osjetilo prisutnosti i pokreta to vrijeme nije definirano. Kroz testiranje karakteristika ovih osjetila pokazalo se da je frekvencija slanja podataka osjetila za vitalne znakove oko 3 sekunde za nova mjerenja dok je za osjetilo prisutnosti to vrijeme kraće i iznosi 1 sekundu.

Struktura konfiguracije osjetila spojenog na ESP32 mikrokontroler

Programski kod ESP32 sastoji se od nekoliko značajki. Povezivanje na WiFi mrežu omogućuje uspostavljanje veze ESP32 na lokalnu bežičnu mrežu. Čitanje podataka s osjetila ostvareno je korištenjem knjižnice za komunikaciju s osjetilom isporučene od strane proizvođača osjetila. Prikupljeni podaci zatim se formatiraju u JSON strukturu prikladnu za slanje putem HTTP zahtjeva. Na kraju, slanje podataka na poslužitelj ostvareno je slanjem HTTP POST zahtjeva koji prenosi očitavanja s osjetila na centralno računalo.

Sljedeći isječak koda prikazuje osnovne konfiguracijske parametre koji omogućuju:

- spajanje ESP32 mikrokontrolera na WiFi mrežu,
- odabir identifikatora osjetila te njegovog tipa,
- adresu centralnog računala na koje se mjerenja šalju.

```
1 const char *WIFI_SSID = "*****";           // Replace with your WiFi  
   SSID  
2 const char *WIFI_PASSWORD = "*****";     // Replace with your WiFi  
   password
```

```

3 const char *SERVER_ENDPOINT = ".../send_data"; // Replace with your server
    endpoint, it can be obtained by copying the link upon running the
    server
4 const String SENSOR_ID = "HA_live_001"; // Replace with your sensor
    ID
5 const String SENSOR_TYPE = "heart_and_breath"; // Replace with your sensor
    type

```

Isječak kôda 7..15: Konfiguracija za ESP32 mikrokontroler potrebna za spajanje na WiFi mrežu

Povezivanje na WiFi

Kako bi ESP32 mogao komunicirati s poslužiteljem, najprije se uspostavlja WiFi veza:

```

1 // Function to setup WiFi connection
2 void setupWiFi()
3 {
4   WiFi.begin(WIFI_SSID, WIFI_PASSWORD); // Start WiFi connection with
    provided SSID and password
5
6   Serial.print("Connecting to WiFi");
7   while (WiFi.status() != WL_CONNECTED) // Wait until the device is
    connected to WiFi
8   {
9     delay(500); // Wait for 500 milliseconds
10    Serial.print("."); // Print a dot for each attempt to connect
11  }
12
13  Serial.println("\nWiFi connected"); // Print message when connected
14  Serial.println(WiFi.localIP()); // Print the local IP address assigned
    to the device
15 }

```

Isječak kôda 7..16: Konfiguracija za ESP32 mikrokontroler potrebna za spajanje na WiFi mrežu

ESP32 se kontinuirano pokušava povezati na mrežu sve dok veza ne bude uspostavljena, a nakon toga ispisuje dodijeljenu IP adresu.

Generiranje JSON formata za slanje podataka

Podaci očitani s osjetila formatiraju se u JSON strukturu prije nego što se pošalju poslužitelju:

```
1
2 // Function to create a JSON payload with sensor data
3 String createJsonPayload(int heartRate, int breathRate)
4 {
5     String payload = "{";
6     payload += "\"sensor_id\": \"" + SENSOR_ID + "\", "; // Add sensor ID to
7     // the payload
8     payload += "\"type\": \"" + SENSOR_TYPE + "\", "; // Add sensor type to
9     // the payload
10    payload += "\"time\": " + String(millis()) + ", "; // Add current time (
11    // in milliseconds) to the payload
12
13    // Add heart rate to the payload, or "-" if heart rate is not available
14    if (heartRate != -1)
15    {
16        payload += "\"heart_rate\": " + String(heartRate) + ", ";
17    }
18    else
19    {
20        payload += "\"heart_rate\": \"-\", ";
21    }
22
23    // Add breath rate to the payload, or "-" if breath rate is not
24    // available
25    if (breathRate != -1)
26    {
27        payload += "\"breath_rate\": " + String(breathRate);
28    }
29    payload += "}";
30}
```



```

25 else
26 {
27     payload += "\"breath_rate\":\\"-\"";
28 }
29
30 payload += "}"; // Close the JSON object
31 return payload; // Return the JSON payload as a string
32 }

```

Isječak kôda 7..17: Funkcija koja stvara JSON format prilikom slanja podataka s osjetila na poslužitelj

Ako su podaci s osjetila dostupni, oni se dodaju u JSON format, dok se u suprotnom umjesto vrijednosti postavlja znak "-" koji označava nedostatak podataka.

Slanje podataka na poslužitelj

ESP32 mikrokontroler šalje podatke s osjetila poslužitelju s pomoću HTTP POST zahtjeva:

```

1 bool sendSensorData(int heartRate, int breathRate)
2 {
3     // Check if the WiFi is connected, if not, reconnect
4     if (WiFi.status() != WL_CONNECTED)
5     {
6         setupWiFi();
7     }
8
9     // Initialize HTTP client
10    HTTPClient http;
11    http.begin(SERVER_ENDPOINT);
12    http.addHeader("Content-Type", "application/json");
13
14    // Create JSON payload with sensor data
15    String payload = createJsonPayload(heartRate, breathRate);

```

```

16 Serial.println("Sending payload: " + payload);
17
18 // Send HTTP POST request with the payload
19 int httpResponseCode = http.POST(payload);
20
21 // Print HTTP response code
22 Serial.print("HTTP Response code: ");
23 Serial.println(httpResponseCode);
24
25 // If the response code is positive, print the server response
26 if (httpResponseCode > 0)
27 {
28     String response = http.getString();
29     Serial.println("Server response: " + response);
30 }
31 else
32 {
33     // Print error message if the POST request failed
34     Serial.printf("Error on sending POST: %s\n", http.errorToString(
35         httpResponseCode).c_str());
36 }
37 // End the HTTP connection
38 http.end();
39
40 // Return true if the response code is positive, otherwise false
41 return httpResponseCode > 0;
42 }

```

Isječak kôda 7..18: Funkcija za slanje mjerenja s osjetila na polužitelj

Primjer jednog HTTP zahtjeva sa pripadajućom JSON porukom koja uključuje podatke o osjetilu i njegovim trenutnim mjerenjima izgleda ovako:

```
POST /send_data HTTP/1.1
```

```
Host: your.server.address
Content-Type: application/json
Content-Length: 113
```

```
{
  "sensor_id": "HA_live_001",
  "type": "heart_and_breath",
  "time": 20240,
  "heart_rate": 72,
  "breath_rate": 13
}
```

Ako WiFi veza nije dostupna, funkcija ponovo pokušava uspostaviti vezu. Podaci se šalju poslužitelju u JSON formatu, a odgovor se ispisuje na serijskom naredbenom retku.

Glavna petlja programa

ESP32 kontinuirano očitava podatke s osjetila i šalje ih poslužitelju:

```
1 void loop()
2 {
3   // Read data from the radar sensor
4   radar.Breath_Heart();
5
6   // Check if the sensor has reported any data
7   if (radar.sensor_report != 0x00)
8   {
9     // Process the sensor report
10    switch (radar.sensor_report)
11    {
12    case HEARTRATEVAL:
13      // Update the last heart rate value
14      lastHeartRate = radar.heart_rate;
15      break;
```

```

16  case BREATHVAL:
17      // Update the last breath rate value
18      lastBreathRate = radar.breath_rate;
19      break;
20  }
21
22  // Send sensor data if there is a valid heart rate or breath rate
23  if (lastHeartRate != -1 || lastBreathRate != -1)
24  {
25      sendSensorData(lastHeartRate, lastBreathRate);
26  }
27  }
28
29  // Delay to avoid program jamming
30  delay(200);
31  }

```

Isječak kôda 7..19: Glavna petlja koja se izvodi na ESP32 mikrokontroleru

Mjerenja s osjetila se kontinuirano očitavaju, a podaci se šalju samo ako postoji novo očitavanje otkucaja srca ili ritma disanja. Postavljen je kratki vremenski odmak od 200 ms između iteracija kako bi se smanjilo opterećenje sustava, a tako nalaže i sam proizvođač osjetila.

7.8. Testiranje sustava

Sustav je testiran s pomoću simulatora aktivnosti, koji čita unaprijed snimljene podatke s osjetila i šalje ih poslužitelju, te s pomoću frontend sučelja, koje omogućuje jednostavno praćenje rada sustava.

Glavna metoda testiranja sustava bila je korištenje simulatora aktivnosti. Simulator oponaša rad osjetila korištenjem snimljenih podataka pohranjenih u CSV datotekama te ih šalje poslužitelju u vremenskim razmacima koji odgovaraju stvarnim očitanjima s osjetila. Ovaj pristup omogućio je provjeru ispravnog primanja i obrade podataka s osjetila, provjeru ispravnosti praćenja zauzetosti pros-

torija te testiranje prepoznavanja psihofizičkog stanja korisnika.

Frontend sučelje i WebSocket komunikacija

Frontend sučelje omogućuje prikaz mjerenja s osjetila u stvarnom vremenu te uklanja potrebu za ručnim slanjem HTTP zahtjeva kroz naredbeni redak kako bi se vidjele promjene u sustavu. WebSocket komunikacija omogućuje dvosmjernu vezu između poslužitelja i klijenta, čime se neutralizira kašnjenje i osigurava trenutno ažuriranje podataka.

Na strani poslužitelja, WebSocket veza postavlja se kako bi se svi povezani klijenti mogli obavještavati o promjenama u sustavu:

```
1 @app.websocket("/ws")
2 async def websocket_endpoint(websocket: WebSocket):
3     # Accept the WebSocket connection
4     await websocket.accept()
5     # Add the connected client to the set of connected clients
6     connected_clients.add(websocket)
7     try:
8         while True:
9             # Wait to receive data from the client
10            data = await websocket.receive_text()
11            print(f"Received from client: {data}")
12        except WebSocketDisconnect:
13            # Remove the client from the set of connected clients when they
14            ↪ disconnect
15            connected_clients.remove(websocket)
```

Isječak kôda 7..20: Metoda uspostave WebSocket veza između poslužitelja i klijenata

Kada se uoči promjena u mjerenjima s osjetila na poslužitelju, ažurirane informacije automatski se šalju svim povezanim klijentima:

```
1 # Function to broadcast messages to all connected clients
2 async def broadcast_update(sensor_id, sensor_data):
```

```

3   # Get the room name from the sensor
4   room_name = room_manager.sensors[sensor_id].room
5
6   # Structure the data to be sent to clients
7   structured_data = {
8       "type": "room_update",
9       "room": room_name,
10      "sensor_id": sensor_id,
11      "sensor_data": sensor_data,
12      "physiological_state": person_manager.physiological_state,
13      "last_known_room": person_manager.last_known_room
14  }
15
16  # List to keep track of clients that disconnect during the broadcast
17  disconnected_clients = []
18
19  # Send the structured data to all connected clients
20  for client in connected_clients:
21      try:
22          await client.send_text(json.dumps(structured_data))
23      except WebSocketDisconnect:
24          # If a client disconnects, add it to the list of disconnected
25             ↪ clients
26          disconnected_clients.append(client)
27
28  # Remove all disconnected clients from the set of connected clients
29  for client in disconnected_clients:
30      connected_clients.remove(client)

```

Isječak kôda 7.21: Funkcija odašiljanja informacija o ažuriranom stanju sustava

Na frontend sučelju, WebSocket veza omogućuje primanje ažuriranih podataka s osjetila:

```

1 // Establish a WebSocket connection to the backend server
2 const socket = new WebSocket("ws://localhost:8000/ws");

```

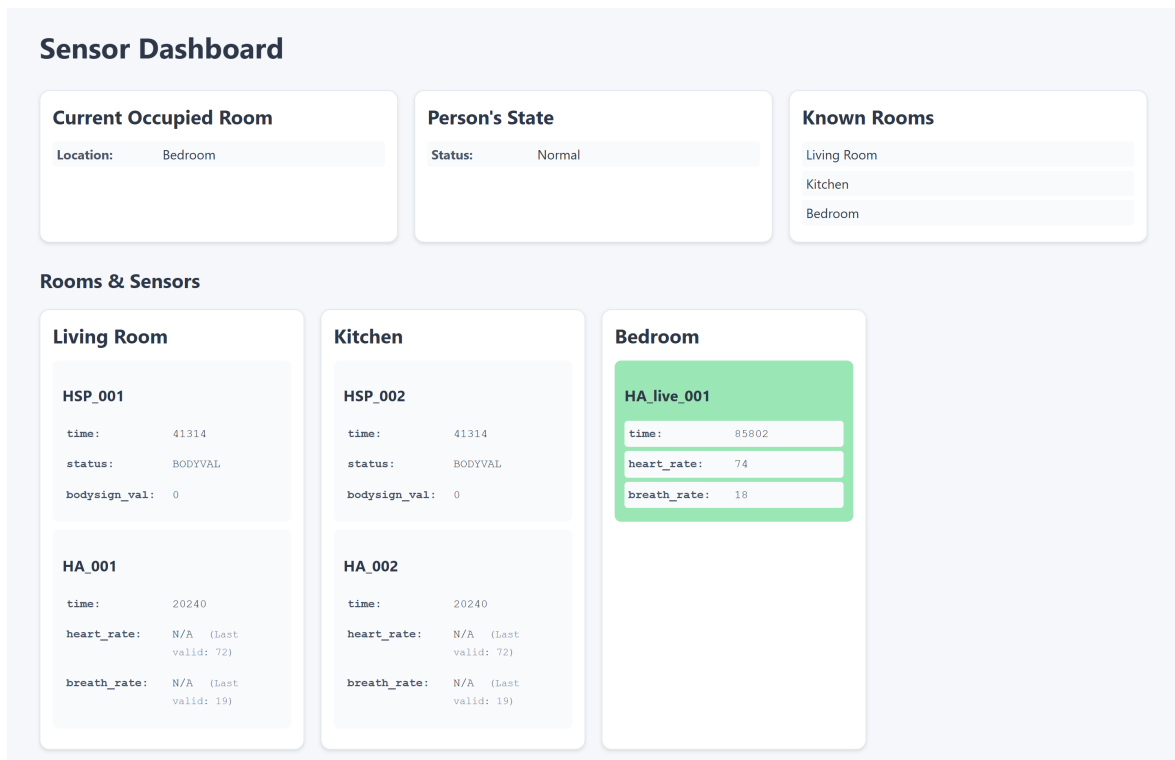
```

3
4 ...
5
6 // Handle incoming WebSocket messages
7 socket.onmessage = (event) => {
8     const message = JSON.parse(event.data);
9
10    // Check if the message is a room update
11    if (message.type === "room_update") {
12        // Update the sensor reading for the specified room and sensor
13        updateSensorReading(message.room, message.sensor_id,
14                               ↪ message.sensor_data);
15
16        // Update the physiological state if provided in the message
17        if (message.physiological_state) {
18            document.getElementById("physiologicalState").innerHTML = '
19                <span class="data-label">Status:</span>
20                <span class="data-value">${message.physiological_state}</span>
21            ';
22        }
23
24        // Update the last known room if provided in the message
25        if (message.last_known_room) {
26            document.getElementById("currentOccupiedRoom").innerHTML = '
27                <span class="data-label">Location:</span>
28                <span class="data-value">${message.last_known_room}</span>
29            ';
30        }
31    };

```

Isječak kôda 7..22: Funkcija uspostave WebSocket veze s poslužiteljem na strani klijenta

Prikaz ovog web sučelja moguće je vidjeti na sljedećoj slici 7.1.:



Slika 7.1. Prikaz web sučelja korištenog za testiranje

Na snimci zaslona ovog web sučelja moguće je uočiti i rad simulatora aktivnosti i podršku sustava za stvarna, fizička osjetila. Kao što je moguće vidjeti na slici, okvir u kojem se nalaze mjerenja pojedinog osjetila u trenutku pristizanja novih podataka na trenutak poprima zelenu pozadinu kako bi se lakše primijetila promjena u sustavu. Također osjetila su sa svojim pripadajućim prostorijama odvojeni u zasebne okvire, dok su informacije o trenutačno zauzetoj prostoriji, trenutačnom stanju korisnika te popisu poznatih prostorija pri vrhu ekrana. Vrijeme koje je prikazano unutar svakog okvira osjetila izraženo je u milisekundama, a označava vrijeme proteklo od početka rada tog osjetila. To se vrijeme u budućem radu naravno može zamijeniti i s vremenskom oznakom trenutka u kojem je mjerenje s osjetila pristiglo na centralno računalo sustava.

Testiranje sustava pokazalo je da poslužitelj uspješno prima i obrađuje mjerenja pristigla s osjetila, dok frontend sučelje omogućuje jednostavno praćenje rada sustava bez potrebe za ručnim slanjem HTTP zahtjeva kroz naredbeni redak. WebSocket komunikacija osigurala je da se podaci ažuriraju trenutno, čime je testiranje postalo točnije i učinkovitije. Simulator aktivnosti omogućio je pro-

vjeru ispravnosti sustava u različitim slučajevima, osiguravajući njegov ispravan rad prije implementacije s fizičkim osjetilima. Osim putem simulatora aktivnosti, sustav je uspješno testiran i s fizičkim osjetilima povezanima na ESP32 mikrokontroler. Time je potvrđena funkcionalnost sustava da u stvarnom vremenu prima i obrađuje podatke sa stvarnih osjetila.

8. Zaključak

U ovom radu predstavljeno je rješenje koje omogućuje nadzor aktivnosti u kućanstvu, s posebnim naglaskom na starije osobe koje žive samostalno. Polazna motivacija proizašla je iz potrebe da se na neinvazivan način kontinuirano prate kretanje, obrasci ponašanja i vitalni znakovi korisnika, što je od velike važnosti za pravovremenu detekciju kritičnih situacija i njihovo sprječavanje. Posebno su istaknuti demografski trendovi koji ukazuju na rastući broj starijih, čime se naglašava potreba za ovakvim vrstama tehnoloških rješenja za stariju populaciju koja živi samostalno.

Temelj sustava čine mmWave osjetila za praćenje prisutnosti i pokreta te osjetila za mjerenje vitalnih znakova (otkucaja srca i ritma disanja). Ta osjetila ne narušavaju privatnost korisnika, a istodobno pružaju točne informacije o kretanju i vitalnim znakovima korisnika. U radu je pokazano da je, uz odgovarajuću konfiguraciju i postavljanje osjetila, moguće postići zadovoljavajuću točnost prepoznavanja aktivnosti te pravodobnu detekciju potencijalno opasnih stanja. Provedena ispitivanja ukazala su na važnost pravilnog odabira lokacije i kuta pod kojim se osjetila ugrađuju, a također je naglašena potreba za detaljnim testiranjima u stvarnim kućnim uvjetima.

Razvijeno središnje računalo (poslužitelj) pokazalo je kako se prikupljeni podaci mogu uspješno obraditi u stvarnom vremenu, dok je frontend sučelje omogućilo jednostavan prikaz informacija o zauzetosti prostorija i psihofizičkom stanju korisnika. Jedna od većih vrijednosti sustava jest i simulator aktivnosti koji reproducira ponašanje osjetila na temelju stvarnih snimaka mjerenja, što omogućuje testiranje funkcionalnosti i pouzdanosti sustava bez potrebe za fizičkim osjetilima u svakom trenutku.

Modularnost i prilagodljivost sustava vidljivi su u jednostavnoj integraciji dodatnih vrsta osjetila ili povećanju broja postojećih. Time je omogućen skalabilan pristup, koji se može prilagoditi različitim tipovima kućanstava i individualnim potrebama korisnika. Sustav tako nudi širok potencijal primjene, od praćenja osnovnih aktivnosti do složenijih scenarija u kojima se nadziru obrasci ponašanja i vitalni znakovi u većem opsegu.

Iako je bilo predviđeno uključiti i osjetilo za detekciju pada, ono u konačnici nije uspješno integrirano u sustav. Uspostavilo se da postoje poteškoće ili neispravnost samog osjetila, što zahtijeva dodatna testiranja i potencijalnu zamjenu sklopovlja odnosno osjetila. U budućem radu, preporučuje se i detaljnija validacija preciznosti mjerenja vitalnih znakova usporedbom s referentnim medicinskim uređajima, kao i izvedba dugotrajnijih testiranja u stvarnim okruženjima. Dodatno, funkcionalnosti sustava mogle bi se poboljšati primjenom algoritama strojnog učenja za napredniju analizu obrazaca ponašanja i ranu detekciju promjena u zdravstvenom stanju korisnika. Time bi se osigurala još bolja kvaliteta sustava.

Literatura

- [1] E. Ahmed, I. Yaqoob, A. Gani, M. Imran, i M. Guizani, "Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges", *IEEE Wireless Communications*, sv. 23, br. 5, str. 10–16, 2016. <https://doi.org/10.1109/MWC.2016.7721736>
- [2] P. F. Edemekong, D. L. Bomgaars, S. Sukumaran, i C. Schoo, *Activities of Daily Living*. StatPearls Publishing, Treasure Island (FL), 2023. [Mrežno]. Adresa: <http://europepmc.org/books/NBK470404>
- [3] K. Viard, "Modelling and Recognition of Human Activities of Daily Living in a Smart Home", Theses, Université Paris Saclay (COMUE) ; Politecnico di Bari. Dipartimento di Ingegneria Elettrica e dell'Informazione (Italia), srpanj 2018. [Mrežno]. Adresa: <https://theses.hal.science/tel-01867623>
- [4] A. Chaaraoui, P. Climent i Pérez, i F. Flórez-Revuelta, "A review on vision techniques applied to human behaviour analysis for ambient-assisted living", *Expert Systems with Applications*, sv. 39, str. 10 873 – 10 888, 09 2012. <https://doi.org/10.1016/j.eswa.2012.03.005>
- [5] A. Fleury, N. Noury, i M. Vacher, "Supervised classification of activities of daily living in health smart homes using svm", *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, sv. 2009, str. 6099–102, 09 2009. <https://doi.org/10.1109/IEMBS.2009.5334931>
- [6] Wikipedija, "Radar — wikipedija, slobodna enciklopedija", 2024.,

- [mrežno; stranica posjećena: prosinac 2024.]. [Mrežno]. Adresa: <https://hr.wikipedia.org/wiki/Radar>
- [7] C. Iovescu i S. Rao, "The fundamentals of millimeter wave sensors", *Texas Instruments*, str. 1–8, 2017.
- [8] Maritime Studies South Africa, [https://maritimesa.org/nautical-science-grade-12/2020/09/09/use-of-marine-radar/#:~:text=Most%20marine%20radars%20operate%20in,S%20band%20\(3%20GHz\).](https://maritimesa.org/nautical-science-grade-12/2020/09/09/use-of-marine-radar/#:~:text=Most%20marine%20radars%20operate%20in,S%20band%20(3%20GHz).), [mrežno; stranica posjećena: prosinac 2024.].
- [9] Team Oxeltech, <https://oxeltech.de/en/a-beginners-guide-to-millimeter-wave-mmwave-radar-technology-applications-and-advantages/>, [mrežno; stranica posjećena: prosinac 2024.].
- [10] Christian Wolff, <https://www.radartutorial.eu/02.basics/Frequency%20Modulated%20Continuous%20Wave%20Radar.en.html>, [mrežno; stranica posjećena: prosinac 2024.].
- [11] Snehal Buche, <https://www.design-reuse.com/articles/55851/mmwave-radar-principle-applications.html#:~:text=mmWave%20RADAR%20is%20a%20non,be%20performed%20with%20no%20interference.>, [mrežno; stranica posjećena: prosinac 2024.].
- [12] Minew, <https://www.iotforall.com/iot-sensors-millimeter-wave-radar-vs-competing-technologies>, [mrežno; stranica posjećena: prosinac 2024.].
- [13] SeeedStudio, https://wiki.seeedstudio.com/Radar_MR24HPC1/, [mrežno; stranica posjećena: studeni 2024.].
- [14] Seeedstudio, https://files.seeedstudio.com/wiki/mmWave-radar/24GHz_mmWave_Sensor-Human_Static_Presence_Module_Lite_Datasheet.pdf, [mrežno; stranica posjećena: studeni 2024.].
- [15] —, https://files.seeedstudio.com/wiki/mmWave-radar/24GHz_mmWave_Sensor-Human_Static_Presence_Module_Lite_Datasheet.pdf, [mrežno; stranica posjećena: studeni 2024.].

- [16] SeeedStudio, https://wiki.seeedstudio.com/Radar_MR60BHA1/, [mrežno; stranica posjećena: studeni 2024.].
- [17] Seeedstudio, https://files.seeedstudio.com/wiki/mmWave-radar/MR60BHA1_Datasheet.pdf, [mrežno; stranica posjećena: studeni 2024.].
- [18] —, https://files.seeedstudio.com/wiki/60GHzradar/new_res/MR60BHA1_userManual_V2.3.pdf, [mrežno; stranica posjećena: studeni 2024.].
- [19] SeeedStudio, https://wiki.seeedstudio.com/Radar_MR60FDA1/, [mrežno; stranica posjećena: studeni 2024.].
- [20] Seeedstudio, https://files.seeedstudio.com/wiki/60GHzradar/new_res/MR60FDA1_user_manual-V1.3.pdf, [mrežno; stranica posjećena: studeni 2024.].
- [21] —, https://files.seeedstudio.com/wiki/60GHzradar/60GHz_mmWave_Sensor_Fall_Detection_Module_Pro-Universal_Protocol.pdf, [mrežno; stranica posjećena: studeni 2024.].

Sažetak

Sustav za prepoznavanje aktivnosti osobe u kućanstvu

Mislav Kovačić

Ovaj rad bavi se razvojem sustava za nadzor aktivnosti starijih osoba koje žive samostalno, uz primjenu neinvazivnih mmWave osjetila. Glavni cilj je detekcija prisutnosti, pokreta i vitalnih znakova (otkucaja srca i ritma disanja), čime se omogućuje brzo uočavanje potencijalno opasnih situacija u kućanstvu. Istražene su tehničke i društvene prepreke uvođenju takvih rješenja, posebno u pogledu privatnosti i prihvaćenosti od strane korisnika. Predstavljeno je središnje računalo za obradu podataka u stvarnom vremenu, frontend sučelje za pregled informacija te simulator aktivnosti za testiranje bez fizičkih osjetila. Sustav nudi modularnost i mogućnost daljnjeg proširenja, primjerice primjenom algoritama strojnog učenja, čime bi se osigurala sveobuhvatnija zaštita i poboljšana kvaliteta praćenja aktivnosti i zdravstvenog stanja korisnika.

Ključne riječi: pametna kućanstva; starije osobe; mmWave; mmWave osjetila; osjetila; praćenje aktivnosti; detekcija pokreta; vitalni znakovi; nadzor; neinvazivni nadzor; privatnost; Internet stvari; IoT

Abstract

A household activity recognition system for individuals

Mislav Kovačić

This paper deals with the development of a system for monitoring the activities of older adults living independently, utilizing non-invasive mmWave sensors. The primary goal is to detect presence, movement, and vital signs (heart rate and breathing rate), thereby enabling the timely detection of potentially hazardous situations in the home environment. Technical and social challenges in introducing such solutions—particularly those related to privacy and user acceptance—have been explored. The paper presents a central computer for real-time data processing, a frontend interface for viewing information, and an activity simulator for testing without the need for physical sensors. The system offers modularity and the possibility of further expansion, for instance through the application of machine learning algorithms, thus providing more comprehensive protection and improved quality in monitoring users' activities and health status.

Keywords: smart homes; elderly; mmWave; mmWave sensors; sensors; activity monitoring; motion detection; vital signs; monitoring; non-invasive monitoring; privacy; Internet of Things; IoT