Razvoj sustava za prikupljanje podataka, praćenje i analizu zdravstvenog stanja pacijenta

Brekalo, Bruno

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:851318

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-17**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Sveučilišni studij

Razvoj sustava za prikupljanje podataka, praćenje i analizu zdravstvenog stanja pacijenta

Diplomski rad

Bruno Brekalo

Osijek, 2019.

Sadržaj

1. UVOD ... 1

2. PRIKUPLJANJE BIOMEDICINSKIH PODATAKA ... 2

2.1 Elektrokardiogram ... 2

2.2 Tjelesna temperatura ... 2

2.3 Krvni tlak ... 3

2.4 Elektrodermalna aktivnost ... 3

2.5 Elektromiografija ... 4

2.6 Pulsna oksimetrija ... 4

3. PRIKAZ POSTOJEĆIH RJEŠENJA I IDEJNO RJEŠENJE VLASTITOG SUSTAVA ... 6

3.1 Zahtjevi na prikupljanje podataka ... 6

3.2 Postojeća rješenja ... 6

3.2.1 Philips IntelliVue X3 ... 7

3.2.2 GE CARESCAPE Monitor B850 ... 8

3.2.3 Philips IntelliVue MX40 ... 8

3.2.4 Surveyor S4 ... 9

3.3 Idejno rješenje sustava za prikupljanje podataka, praćenje i analizu zdravstvenog stanja pacijenta ... 9

4. PROGRAMSKO OSTVARENJE SUSTAVA ZA PRIKUPLJANJE I ANALIZU BIOMEDICINSKIH PODATAKA PACIJENTA ... 12

4.1 Opis ostvarenja sustava ... 12

4.2 Raspberry Pi ... 13

4.3 Arduino Leonardo ... 14

4.4 E-Health senzorska platforma ... 14

4.5 Primijenjena metoda strojnog učenja ... 15

4.5.1 Strojno učenje ... 15

4.5.2 Metoda K-najbližih susjeda ... 16

4.5.3 Prikaz algoritma na Raspberry Pi-u ... 17

4.6 Mjerenje EKG-a ... 17

4.7 Mjerenje tjelesne temperature ... 19

4.8 Mjerenje krvnog tlaka ... 21

4.9 Mjerenje elektrodermalne aktivnosti ... 21

4.10 Elektromiografija ... 23

4.11 Puls i zasićenost krvi kisikom ... 25

5. PRIKAZ RADA SUSTAVA U STVARNIM UVJETIMA S REZULTATIMA I ANALIZOM 27

5.1 Mjerenje tjelesne temperature ... 27

5.2 Mjerenje zasićenosti krvi kisikom i broja otkucaja srca u minuti 29
5.3 Izrada EKG-a 30
5.4 Izrada EMG-a 31
5.5 Izrada grafa elektrodermalne aktivnosti 32
5.6 Mjerenje krvnog tlaka 34
5.7 Svrha strojnog učenja 35
5.8 Prikaz dobivenih rezultata i način spremanja 35
5.9 Analiza rada sustava u stvarnim uvjetima 36
6. ZAKLJUČAK 38
7. LITERATURA 39
SAŽETAK 41
ABSTRACT 42
ŽIVOTOPIS 43
PRILOZI 44
Prilog 1. Programski kod programa na Raspberry Pi-u 44
Prilog 2. Programski kod programa na Arduinu 47

1. UVOD

Prikupljanje podataka o zdravstvenom stanju pacijenta korisno je, osim za određivanje zdravstvenog stanja, za dijagnosticiranje bolesti. Prikupljanje podataka obavlja se uz pomoć senzora koji se spajaju na pacijentovo tijelo. Ti senzori mjere određene biomedicinske podatke poput tjelesne temperature, krvnog tlaka i slično. Pomoću tih podataka medicinsko osoblje može odrediti kakvo je zdravstveno stanje pacijenta.

Zadatak ovog diplomskog rada je razvoj sustava za prikupljanje podataka, praćenje i analizu zdravstvenog stanja pacijenta pomoću računala Raspberry Pi i mikroupravljača Arduino. Od gotovih rješenja, osim Arduina i Raspberry Pi, korištena je senzorska platforma *e-health* koja omogućuje prikupljanje podataka sa senzora te pripadajući senzori. Za očitanje podataka sa senzora potrebna je biblioteka Ehealth.h koju Libelium pruža uz svoje senzorske platforme *e-health*. U radu je Arduino korišten uz senzorsku platformu *e-health* za prikupljanje pacijentovih podataka sa senzora te ih šalje na računalo Raspberry Pi koje provodi obradu podataka, primjenjuje strojno učenje na neke od podataka te obavještava pacijenta o njegovom stanju, odnosno na zaslon prikazuje izmjerene podatke sa senzora. Obrada podataka ima za cilj utvrditi odstupanja podataka od medicinski određenih granica. Strojno učenje primijenjeno je na pacijentovu tjelesnu temperaturu, te broj otkucaja srca u minuti. Ti podaci često znaju odstupati od medicinskih granica iako pacijent nije bolestan, tako da je primjenom strojnog učenja moguće za svakog pacijenta odrediti granice vrijednosti, odnosno utvrditi je li pacijent bolestan ili ne bez obzira na odstupanja.

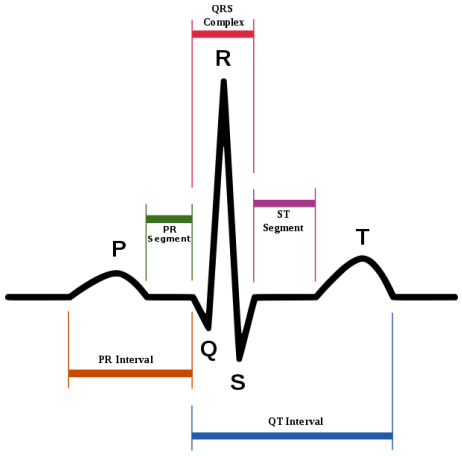
U poglavlju 2 dana je teorijska podloga prikupljanja potrebna za razumijevanje prikupljanja i obradbe raznih biomedicinskih podataka, te njihovih raspona vrijednosti. Poglavlje 3 bavi se prikazom sličnih postojećih rješenja i predlaganje vlastitog rješenja. U poglavlju 4 opisano je sklopovsko i programsko ostvarenje sustava, te detaljni opisi svakog njegovog dijela, načina rada, oblici podataka i rezultata. U poglavlju 5 prikazan je način korištenja sustava, te primjeri primjene u stvarnim uvjetima. Prikupljeni su, prikazani i analizirani rezultati dobivenim mjerenjima.

2. PRIKUPLJANJE BIOMEDICINSKIH PODATAKA

Medicinska dijagnostika je vrlo važan aspekt medicine. Dijagnostika obuhvaća načine identifikacije određenih problema kod ljudi poput bolesti, nepravilnosti kosti i tkiva, organske promjene i slično, a sve to uz pomoć računala. Prilikom dijagnosticiranja, prvi korak je prikupljanje potrebnih biomedicinskih podataka koristeći medicinskom osoblju dostupne uređaje [1]. Najčešće korišteni uređaji i dijagnostičke metode su opisane u nastavku rada.

2.1 Elektrokardiogram

Elektrokardiogram je dijagnostička metoda koji se rutinski koristi za procjenu električke i mišićne funkcije srca. Analizom valova i vektora depolarizacije i repolarizacije moguće je dijagnosticirati bolesti. EKG funkcionira tako da srcem prolazi srčani impuls koji se širi u tkivo oko srca. Jedan dio tih struja dolazi do kože, gdje nastaju razlike potencijala, koje detektiraju elektrode na koži. Elektrokardiogram bilježi 12 odvoda koji predstavljaju razliku električnih potencijala između elektroda na tijelu. Odvode se dijele u dvije grupe, periferne odvode i prekordijalne odvode. EKG zdrave osobe sadrži P-val, QRS-kompleks (kompleks je sastavljen od Q, R i S vala) i T-val (Sl. 2.1) [2].



Sl. 2.1 *Shematski prikaz normalnog EKG-a [2]*

2.2 Tjelesna temperatura

Tjelesna temperatura je stupanj zagrijanosti organizma, ljudi ili životinja. Toplina se u organizmu stvara mišićnim radom, asimilacijom hrane i životnim procesima koji doprinose bazalnom metabolizmu, a gubi se radijacijom i vaporizacijom vode iz dišnih putova i kože.

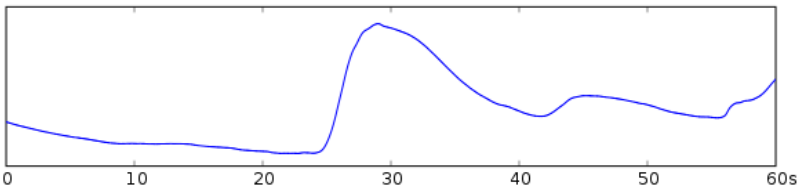
Normalna tjelesna temperatura čovjeka kreće se u rasponu od 36,3 °C – 37,1 °C. Temperatura raspona od 37,1°C do 37,8 °C predstavlja temperaturu niskog stupnja. Porast tjelesne temperature u rasponu od 38°C do 39°C smatra se povišenom tjelesnom temperaturom. Daljnji porast u rasponu od 39°C do 40°C smatra se znatno povišenom tjelesnom temperaturom a temperatura u rasponu od 40°C i 40,5°C smatra se visokom temperaturom. Iako je povišena tjelesna temperatura jedan od indikatora bolesti čovjeka, samo povišenje tjelesne temperature nije dovoljno za procjenu vjerojatnosti postojanja infekcije niti njezinog uzroka. Pri mjerenju tjelesne temperature najčešće se koristi digitalni toplomjer [3].

2.3 Krvni tlak

Krvni tlak je tlak kojim krv djeluje na stjenke krvnih žila u tijelu. Samo kretanje krvi kroz žile posljedica je toga što se krv nalazi pod određenim tlakom. Rad srca rezultira tlakom, to jest pri radu srca, tijekom izbacivanja krvi iz srca tlak raste (sistola), a pri ulijevanju krvi u srce tlak opada (dijastola). Mjerenje tlaka sastoji se od dvije vrijednosti, gornje vrijednosti (sistolički) te donje vrijednosti (dijastolički). Krvni tlak zdrave osobe prosječno je oko 120/80 milimetara stupca žive (mmHg). Starenjem te vrijednosti rastu, no ne prelaze granicu od 140/90 mmHg što je ujedno i granična vrijednost normalnog krvnog tlaka. Visoki krvni tlak se naziva hipertenzija, te je glavni uzročnik razvoja moždanog i srčanog udara, aneurizmi, zatajenja srca te oštećenja bubrega. Niski krvni tlak se naziva hipotenzija i dovodi do vrtoglavica, malaksalosti, zamagljenog vida, opće slabosti ili nesvjestice [4].

2.4 Elektrodermalna aktivnost

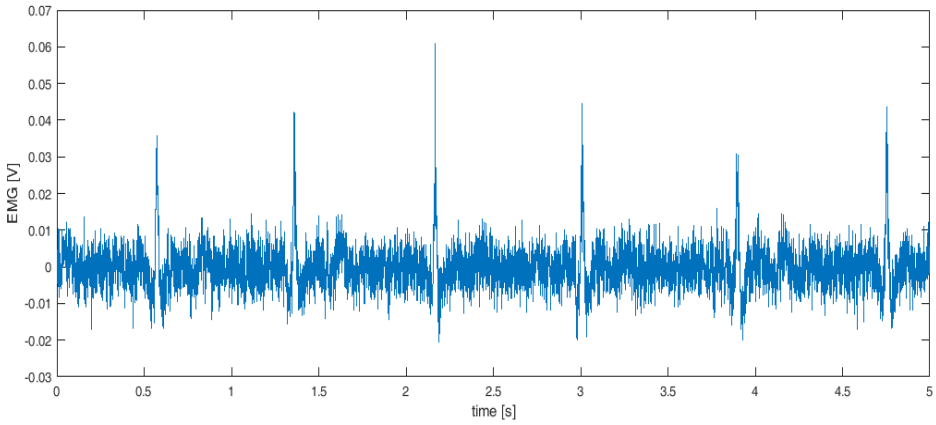
Elektrodermalna aktivnost je svojstvo ljudskog tijela koje uzrokuje varijacije u električnim karakteristikama kože. Električna otpornost kože varira u ovisnosti o stanju znojnih žlijezda u tijelu. Simpatički živčani sustav je odgovoran za upravljanje znojnih žlijezda i samim time znojenjem tijela. Ako je simpatički živčani sustav pod nekim podražajem tada se aktivnost znojnih žlijezda povećava što ujedno povećava električnu vodljivost tijela. Prema tome moguće je mjerenjem električne vodljivosti tijela odrediti emocije koje pojedinac osjeća. Često se električna vodljivost kože kombinira sa zasićenosti kisikom, brojem otkucajima srca i krvnim tlakom kako bi se preciznije odredila dijagnoza. Mjerenje elektrodermalne aktivnosti je jedna od glavnih komponenta modernog poligrafa [5]. Primjer grafa elektrodermalne aktivnosti prikazan je na slici 2.2



Sl. 2.2 *Graf elektrodermalne aktivnosti [5]*

2.5 Elektromiografija

Elektromiografija je dijagnostička metoda za ispitivanje funkcije perifernog živčanog sustava u stanju mirovanja i pokreta. Kada je mišić u mirovanju on ne stvara mišićnu aktivnost, no svako stezanje mišića, bez obzira kako lagano, proizvodi električnu aktivnost, čija vrijednost raste ovisno o jačini stezanja mišića. Proučavanjem živčane provodljivosti mjerimo brzinu kojom motorički živac provodi impulse. Motorički živac potaknut malim električnim nabojem stvara impuls. Stezanje mišića uzrokovano je impulsom koji putuje uzduž živca sve do mišića. Brzina tog impulsa računa se mjerenjem vremena potrebnog impulsu za dosezanje mišića. Mjerenjem te brzine liječnik može dijagnosticirati bolesti mišića te neke neurološke bolesti čiji su simptomi mišićna slabost [6]. Primjer elektromiograma prikazan je na slici 2.3



Sl. 2.3 *Elektromiogram [6]*

2.6 Pulsna oksimetrija

Pulsna oksimetrija je neinvazivna metoda mjerenja zasićenosti krvi kisikom to jest postotka hemoglobina koji je zasićen kisikom. Ljudsko tijelo zahtjeva određen postotak kisika u krvi za normalno funkcioniranje. Najčešće je to udio između 95% i 100%. Ako se udio kisika spusti

ispod 90% može doći do hipoksije. Hipoksija je većinom uzrokovana kroničnim plućnim bolestima, cirkulacijskim poremećajima, popuštanjem srca i drugim čimbenici. Manjak kisika u organizmu prezentira se simptomima poput osjećaja umora, hladnoće, smanjene sposobnosti pamćenja i koncentracije, bolova u mišićima, lupanjem srca, vrtoglavicom te razdražljivošću [7].

Sl. 2.4 *Oksimetar [7]*

3. PRIKAZ POSTOJEĆIH RJEŠENJA I IDEJNO RJEŠENJE VLASTITOG SUSTAVA

Prije izrade vlastitog sustava za prikupljanje i analizu biomedicinskih podataka, potrebno je saznati kakva već rješenja postoje te na koji način rade. Također je potrebno znati na koje zahtjeve za prikupljanje podataka sustav treba odgovoriti to jest koju zadaću treba ispuniti. Idejno rješenje sustava temelji se na već postojećim uređajima, no nadograđuje funkcionalnost postojećih uređaja primjenom strojnog učenja te samom jednostavnošću korištenja.

3.1 Zahtjevi na prikupljanje podataka

Za prikupljanje biomedicinskih podataka potreban je velik broj senzora, a samim time i veliki broj uređaja. Kako bi se olakšalo prikupljanje podataka, razvijeni su monitori pacijenata. Monitori pacijenata su medicinski uređaji koji služe za prikupljanje biomedicinskih podataka. Sastoje se od jednog ili više senzora za prikupljanje podataka, procesorske jedinice za obradu podataka i zaslona. Senzori se spajaju na monitor u svoj odgovarajući priključak i ovisno o pacijentu moguće je prikupljati potrebne podatke, to jest pratiti zdravstveno stanje pacijenta. Monitori pacijenata najčešće prikupljaju podatke poput krvnog tlaka, pulsa i zasićenosti krvi kisikom. Dodavanjem dodatnih senzora na monitor moguće je izraditi EKG, EMG, pratiti tjelesnu temperaturu pacijenta, razinu glukoze i slično. Procesorska jedinica prikuplja podatke sa senzora te radi odgovarajuću obradu podataka kako bi podaci bili razumljivi medicinskom osoblju. Zatim se obrađeni podaci šalju na zaslon te obavještavaju medicinsko osoblje ako neki od prikupljenih podataka odstupa od normalnih granica funkcioniranja ljudskog tijela. Neki monitori također mogu biti i umreženi što omogućuje medicinskom osoblju praćenje više monitora pacijenata odjednom [8].

3.2 Postojeća rješenja

Proučavanjem postojećih rješenja zamijećeno je kako postoje dvije vrste uređaja za mjerenje biomedicinskih podataka. Prva vrsta su monitori pacijenata, uređaji koje korisnik ne može nositi, to jest uređaji se nalaze na jednom mjestu, većih su dimenzija te se koriste gotovo isključivo u bolnicama uz nadzor medicinskog stručnjaka. Druga vrsta su nosivi monitori, uređaji manjih dimenzija, koje pacijent može slobodno nositi i premještati po svojoj volji. Koriste se u medicinskim ambulantama ili u privatne svrhe [22].

Monitori pacijenata mjere velik broj biomedicinskih podataka. Najčešće su to brzina otkucaja srca, krvni tlak, EKG, zasićenost kisikom te količina CO2 u krvi. Jedan monitor je dovoljan za mjerenje svih biomedicinskih podataka uz pomoć pripadajućih senzora. Osim monitora potrebno je i medicinsko osoblje za rukovanje istim. Ovakvu vrstu monitora je gotovo nemoguće kupiti za privatne svrhe, prodaju se samo bolnicama i medicinskom osoblju te su iznimno skupi [8].

Nosivi monitori mjere samo neke od biomedicinskih podataka. Najčešće su to brzina otkucaja srca, krvni tlak, EKG te zasićenost kisikom. Također mjerenje tih biomedicinskih podataka pomoću postojećih rješenja često zahtjeva veći broj uređaja. Sva postojeća rješenja koriste prikaz rezultata na zaslon, no neki sustavi ne obavještavaju korisnika ako izmjereni biomedicinski podaci odstupaju od normalnog rada čovjeka. Neke od prepreka pri korištenju postojećih rješenja su cijena i dostupnost. Većina sustava za mjerenje većeg broja biomedicinskih podataka vrlo je skupa i nije lako dostupna u slobodnoj prodaji. U većini slučajeva doktor pacijentu daje neki od uređaja koji pacijent koristi nekoliko dana te mu pacijent vraća uređaj s prikupljenim podacima koje doktor protumači. Najpoznatiji primjer takvog uređaja je 24-satni holter EKG. Ako pacijent odluči sam kupiti neki od uređaja za osobnu uporabu, najveća prepreka su mu cijena i dostupnost. Sama cijena takvih uređaja je vrlo visoka te ih je moguće kupiti samo u malom broju ljekarni ili preko Interneta [22].

Samo prikupljanje podataka vrši se u stvarnom vremenu što je slučaj sa svim postojećim rješenjima. Analiza podataka obavlja se na uređaju, na kojeg su spojeni senzori. Postojeća rješenja ne koriste algoritme strojnog učenja nego imaju strogo definirane granične vrijednosti, koje nije moguće mijenjati, te ako izmjereni biomedicinski podaci pacijenta odstupaju od tih granica sustav će alarmirati medicinsko osoblje, čak i u slučaju da je pacijent zdrav.

3.2.1 Philips IntelliVue X3

Philips IntelliVue X3 (Sl. 3.1) je monitor pacijenata koji služi za mjerenje biomedicinskih podataka pacijenta prilikom boravka u bolnici i pri transportu u bolnicu. Sastavljen je od kemijski otpornih materijala što ga čini otpornim na sve vrste sredstava za dezinfekciju. Također je otporan na udarce i padove od 1 metra. Može mjeriti tjelesnu temperaturu, zasićenost kisikom, količinu CO2 u krvi, krvni tlak, EKG te broj otkucaja srca u minuti [9].

Sl. 3.1 *Philips IntelliVue X3 [9]*

3.2.2 GE CARESCAPE Monitor B850

GE CARESCAPE Monitor B850 (Sl. 3.2) je monitor pacijenata korišten primarno u bolnicama uz pacijentov krevet. Proširiv je s raznim modulima što mu omogućuje široku primjenu na raznim odjelima unutar bolnice. Pomoću modula moguće je konfigurirati sustav za mjerenje točno određenih biomedicinskih podataka koji su potrebni. Bez modula monitor sam po sebi može mjeriti krvni tlak pacijenta, broj otkucaja srca u minuti, zasićenost kisikom te EKG [10].

Sl. 3.2 *GE CARESCAPE Monitor B850 [10]*

3.2.3 Philips IntelliVue MX40

Philips IntelliVue MX40 (Sl. 3.3) je nosivi monitor koji se koristi u ambulantama ili pacijent može posjedovati kući. S njim se pacijent može slobodno kretati i raditi sve normalne aktivnosti. Također je otporan na kemijska sredstva i otporan na padove do 1 metra. Služi za prikupljanje podataka poput zasićenosti kisikom, broja otkucaja srca u minuti i EKG [11].

Sl. 3.3 *Philips IntelliVue MX40 [11]*

3.2.4 Surveyor S4

Surveyor S4 (Sl. 3.4) je nosivi monitor namijenjen za ambulante i osobnu uporabu. Ima mogućnost praćenja dva pacijenta istovremeno. Također omogućuje slobodno kretanje i aktivnosti. Služi za prikupljanje podataka poput zasićenosti kisikom, broja otkucaja srca u minuti i EKG [12].

Sl. 3.4 *Surveyor S4 [12]*

3.3 Idejno rješenje sustava za prikupljanje podataka, praćenje i analizu zdravstvenog stanja pacijenta

Sustav za prikupljanje podataka, praćenje i analizu zdravstvenog stanja pacijenata zamišljen je kao uređaj koji će korisnik moći samostalno spojiti na sebe te pomoću prikaza na zaslon vidjeti svoje biomedicinske podatke. Uz sam prikaz na zaslon sustav će upozoriti korisnika ako neki od njegovih biomedicinskih podataka nisu unutar prihvatljivih granica normalnog čovjekova funkcioniranja. Sustav također ima i sposobnost strojnog učenja, to jest nakon dužeg korištenja sustav bi sam trebao moći prepoznati ako bi neki od korisnikovih biomedicinskih podataka odstupali od njegovih normalnih biomedicinskih podataka čak i kada su ta odstupanja unutar

granica normalnih biomedicinskih podataka čovjeka. Od biomedicinskih podataka sustav bi mjerio tjelesnu temperaturu korisnika, krvni tlak, zasićenost kisikom, broj otkucaja srca u minuti, elektrodermalnu aktivnost te omogućio izradu elektromiograma i elektrokardiograma pomoću prikupljenih podataka sa senzora. Strojno učenje bi bilo primijenjeno na tjelesnu temperaturu, zasićenost kisikom te brzinu otkucaja srca, jer su to najčešći indikatori o narušenosti homeostaze ljudskog tijela. Ako je otkriveno odstupanje sustav će obavijestiti korisnika koji od senzora je detektirao odstupanje, te ga obavijestiti ako treba potražiti medicinsku pomoć. Mjerni parametri i normalne vrijednosti funkcioniranja ljudskog tijela prikazane su u tablici 3.5.

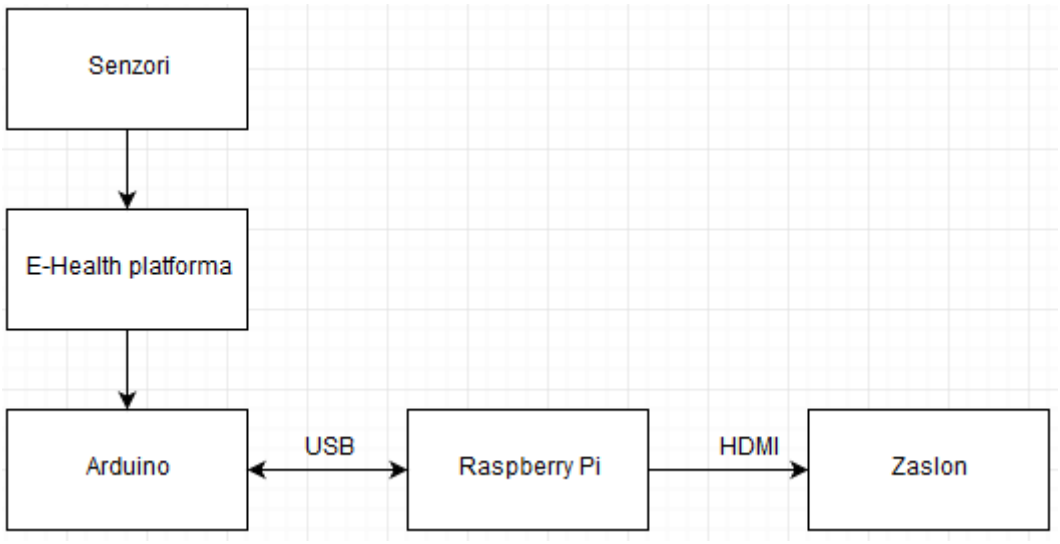
Tablica 3.5. Mjerne vrijednosti

Mjerna vrijednost:	Donja granica:	Gornja granica;
Tjelesna temperatura	36.4°C	37.5°C
Krvni tlak	100/60 mmHg	140/90 mmHg
Zasićenost kisikom	90%	100%
Puls	60	100
Elektrodermalna aktivnost	1μS	4μS
EKG	1mV	5mV
EMG	30μV	30mV

Ako izmjereni biomedicinski podaci odstupaju od vrijednosti iz tablice 3.5 korisnik će biti obaviješten da mu je zdravstveno stanje narušeno. Tjelesna temperatura ispod 36.4°C može se smatrati niskom te upućuje na hipotermiju dok tjelesna temperatura iznad 37.5°C se smatra povišenom temperaturom, koja je česti simptom velikog broja bolesti [3]. Krvni tlak niži od 100/60 mmHg naziva se hipotenzijom koja nastaje kao posljedica zdravstvenih tegoba u organizmu kao što su sepsa, krvarenje, infarkt, trovanje, alergije, teške opekline i dr. Krvni tlak viši od 140/90 mmHg naziva se hipertenzijom te njene uzroke je gotovo nemoguće odrediti zbog brojnosti uzroka poput genetskih predispozicija, faktora okoliša, unosa hrane u organizam i slično [4]. Zasićenost kisikom niža od 90% naziva se hipoksija, čiji uzroci variraju od kroničnih plućnih bolesti poput astme i cistične fibroze, pa sve do cirkulacijskih poremećaja te popuštanja srca [7]. Bradikardija je medicinski naziv za smanjeni broja otkucaja srca, manje od 60 otkucaja u minuti. Bradikardija je najčešće uzrokovana blokadama poput uređenih srčanih problema, no ponekad može nastati kao posljedica lijekova za liječenje visokog tlaka i aritmija. Također se javlja tijekom starenja [13]. Tahikardija je naziv za ubrzani rad srca, to jest više od 100 otkucaja

u minuti. Tahikardija je najčešće uzrokovana reakcijom organizma na neka stanja poput straha ili fizičkog naprezanja. Također se javlja kao posljedica infekcija, upala srca, infarkta miokarda, smanjenog dotoka krvi u srce i slično. [14].

Sustav je osmišljen za izradu koristeći Arduino i Raspberry Pi. Arduino će služiti za prikupljanje podataka sa senzora, preko senzorske platforme te ih prosljeđivati na Raspberry Pi. Raspberry Pi će podatke spremati u tekstualne datoteke te će vršiti analizu primljenih podataka. Primljeni podaci će se uspoređivati prema vrijednostima iz tablice 3.5 te ako je otkriveno odstupanje upozoriti će korisnika ispisom poruke na zaslonu. Također će se primijeniti i metoda strojnog učenja na tjelesnu temperaturu i broj otkucaja srca u minuti. Slika 3.6 prikazuje blok dijagram sustava te korištene protokole za komunikaciju između dijelova sustava.



Sl. 3.6 *Blok dijagram vlastitog sustava*

4. PROGRAMSKO OSTVARENJE SUSTAVA ZA PRIKUPLJANJE I ANALIZU BIOMEDICINSKIH PODATAKA PACIJENTA

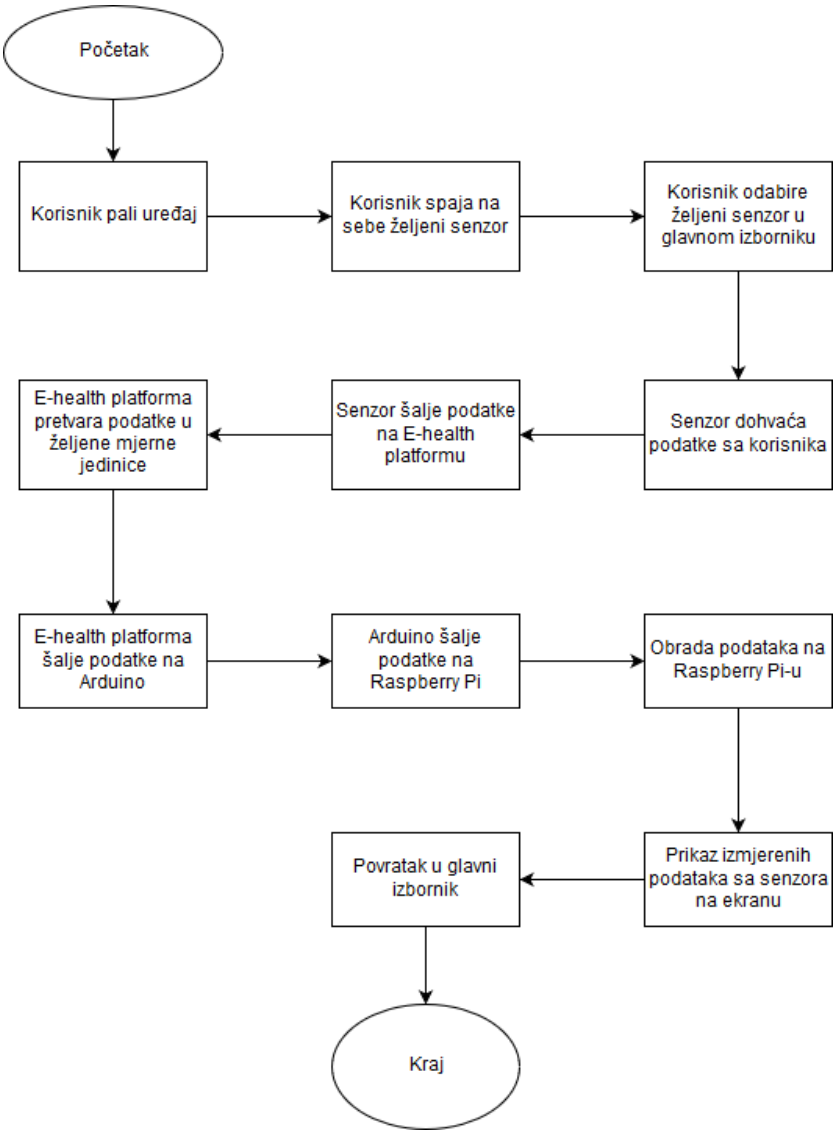
Temeljem idejnog rješenja, sustav je spreman za realizaciju te su odlučeni načini i protokoli komunikacije između raznih dijelova sustava. Idejno rješenje također nudi moguće opcije mikroupravljača za upravljanje sustavom te potrebno računalo za obradu podataka. Obavještavanje korisnika je također adresirano u idejnom rješenju.

4.1 Opis ostvarenja sustava

Sustav je uspješno ostvaren pomoću modula Arduino Leonarda i modula Raspberry Pi. Arduino je korišten za spajanje sa senzorskom platformom *e-health* koja omogućuje povezivanje potrebnih senzora za rad sustava i Arduina. Senzori koji su bili korišteni su senzori za mjerenje EKG-a, senzor za mjerenje tjelesne temperature, senzori za mjerenje elektrodermalne aktivnosti, senzori za EMG, tlakomjer te senzor za mjerenje zasićenosti krvi kisikom i pulsa. Sam Arduino služi za prikupljanje podataka s navedenih senzora te slanje tih podataka na Raspberry Pi koji na sebi ima operacijski sustav Raspbian te obrađuje poslane podatke. Obrada poslanih podataka se temelji na spremanju poslanih podataka te provođenju metode strojnog učenja kako bi sustav sam mogao odrediti jesu li korisnikovi biomedicinski podaci unutar prihvatljivih granica normalnog rada ljudskog sustava. Korištena metoda strojnog učenja je k-najbližih susjeda koja na temelju ulaznih i izlaznih vrijednosti može predvidjeti buduće izlazne vrijednosti pomoću ulaznih, to jest na temelju izmjerenih biomedicinskih podataka može predvidjeti je li korisnik bolestan ili ne. Raspberry također služi i za prikaz rezultata na zaslonu te za obavještavanje korisnika ako neki od prikupljenih biomedicinskih podataka odstupaju od normalnog rada ljudskog sustava.

Programska okolina potrebna za razvoj sustava je Arduino IDE i Spyder. U Arduino IDE napisan je program za prikupljanje podataka sa senzora koristeći programski jezik C. U Spyder-u napisan je program za primanje podataka s Arduina i spremanje te obradu tih podataka koristeći programski jezik Python. Komunikacija između Arduina i Raspberry Pi-a ostvarena je pomoću USB standarda.

Dijagram toka sustava prikazan je na slici 4.1.



Sl. 4.1 *Detaljan opis rada vlastitog sustava*

4.2 Raspberry Pi

Raspberry Pi je računalo smješteno na jednoj pločici veličine kreditne kartice (Sl. 4.2). Na pločici se nalaze sve komponente potrebne za rad nekog računala. Na sebi ima Broadcom BCM2837 sustav na čipu koji sadrži četverojezgreni Cortex-A53 procesor brzine 1.2GHz, VideoCore IV® GPU i 1 GB radne memorije. Raspberry Pi također posjeduje i Ethernet pristup, 4 USB pristupa i 40 GPIO priključaka. Uz Raspberry Pi preporučeno je korištenje nekoga od operacijskih sustava Linux zasnovanog na Debian-u, poput Raspbiana. Zbog svojih mogućnosti i jednostavnosti Raspberry Pi se često koristi u raznim projektima. GPIO priključci mu omogućuju komunikaciju s velikim brojem senzora i uređaja te može obrađivati i prikupljati podatke iz okoline [15].

Sl. 4.2 *Raspberry Pi [15]*

4.3 Arduino Leonardo

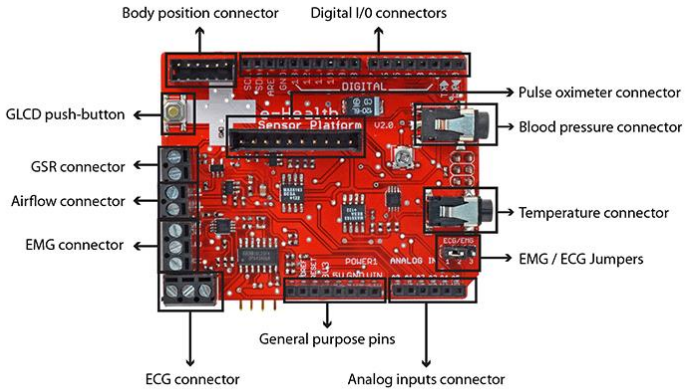
Arduino Leonardo je mikroupravljačka pločica zasnovana na ATmega32u4 mikroupravljaču (Sl. 4.3). Sadrži 20 digitalnih ulazno- izlaznih priključaka od kojih sedam mogu služiti kao naponski izlazi a 12 kao analogni ulazi. Također, sadrži 16MHz kristalni oscilator, micro USB pristup, ulaz za napajanje, ICSP priključke, te prekidač za resetiranje. Zbog brojnosti svojih priključaka te mikroupravljača, Arduino je idealan za povezivanje većeg broja senzora te obradu podataka s tih senzora [16].

Sl. 4.3 *Arduino Leonardo[16]*

4.4 E-Health senzorska platforma

E-Health senzorska platforma (Sl. 4.4) omogućuje korisnicima mjerenje biometrijskih i medicinskih signala kada je potrebno. Pruža podršku za 10 različitih senzora poput: oksimetra, senzora za mjerenje temperature tijela, krvnog tlaka, elektrokardiograma i nekolicinu drugih.

Podaci se mogu koristiti za mjerenje stanja pacijenta u stvarnom vremenu te se mogu bežično slati pomoću Wi-Fi ili Bluetooth protokola. *E-Health* platformu razvila je tvrtka Libelium. Platforma nema medicinske certifikate potrebne za uporabu u medicinske svrhe te za pacijente u kritičnom stanju [17].



Sl. 4.4 *E-Health senzorska platforma [17]*

4.5 Primijenjena metoda strojnog učenja

4.5.1 Strojno učenje

Strojno učenje je grana umjetne inteligencije koja se bavi formiranjem algoritama koji na temelju empirijskih podataka poboljšavaju svoju učinkovitost. Algoritmi strojnog učenja grade matematički model koji se zasniva na ulaznom skupu podataka, zvan trening skup, kako bi algoritam mogao donositi odluke ili predviđanja bez da je izričito programiran za izvođenje tog zadatka. Strojno učenje se može podijeliti na nadzirano učenje, nenadzirano učenje i podržano učenje. U nadziranom učenju algoritam gradi matematički model na osnovu skupa podataka koji sadrži i ulazne podatke i željene izlaze. Cilj nadziranog učenja je predviđanje budućih vrijednosti na temelju ulaznih. Algoritmi s nenadziranim učenjem grade matematički model na osnovu ulaznih podataka bez željenih izlaza. Cilj nenadziranog učenja je pronaći pravilnosti među podacima. Podržano učenje gradi matematički model na temelju izlaznih vrijednosti [18]. Dva su osnovna problema kojim se nadzirano učenje bavi, a to su klasifikacija i regresija. Izlazna varijabla kod klasifikacije je diskretna vrijednost, to jest klasa kojoj ulazni podatak pripada. Kod regresije, izlazna varijabla ima kontinuirane vrijednosti. Zbog svojih izlaznih vrijednosti problem zdravstvenog stanja pacijenta može se svrstati u problem klasifikacije, to jest pacijentovo stanje može pripadati jednoj od dvije klase, zdrav ili bolestan. Ulazni podaci će biti podaci prikupljeni

sa senzora a izlazni podaci će biti pacijentovo zdravstveno stanje. Postoji velik broj algoritama za rješavanje problema klasifikacije. Najčešće se razlikuju po svojoj jednostavnosti primjene, razumljivosti i točnosti. Neke od najčešće korištenih metoda su naivni Bayes klasifikator, Fisherova linearna diskriminanta, logistička regresija, K-najbližih susjeda te perceptron. Zbog svoje jednostavnosti implementacije te točnosti i razumljivosti, metoda koja je odabrana je K-najbližih susjeda [23].

4.5.2 Metoda K-najbližih susjeda

K-najbližih susjeda (k-NN) algoritam je nadzirana vrsta strojnog učenja koja se bazira na određenom broju primjera za koje je unaprijed poznato kojoj klasi pripadaju. Primjeri se sastoje od skupa podataka, to jest svaki primjer sadrži ulazne te njegove pripadajuće izlazne podatke. Za određivanje pripadnosti nekog ulaznog podatka odgovarajućoj klasi, u obzir se uzimaju primjeri *k* klasa čija je vrijednost najbliža ulaznoj. Klasifikacija se određuje temeljem većine, to jest ako većina susjeda pripada određenoj klasi, toj klasi će pripasti i novi podatak [19]. Upravo zbog toga potrebno je postaviti vrijednost *k,* to jest treba odrediti koliko će se susjeda uzeti u obzir prilikom klasificiranja ulaznog podatka. Kako bi se odredila vrijednost k, najčešće se koristi algoritam unakrsne potvrde. Algoritam radi tako da podijeli uzorke u određeni broj segmenata, te se na nekom od segmenata primjenjuje k-NN algoritam s nekom izabranom vrijednosti k, kako bi se odredila točnost. Svakim idućim korakom postupak se ponavlja na nekom drugom segmentu te se na kraju određuje prosječna vrijednost točnosti. Taj algoritam se ponavlja za veći broj vrijednosti k, te nakon usporedbe točnosti, onaj s najvišom se odabire. Zbog same kompleksnosti algoritma i resursa potrebnih, k vrijednost se najčešće odabire proizvoljno [20]. U ovom slučaju, *k* će biti postavljen na vrijednost dva.

Metoda za izračun udaljenosti između nepoznatog uzorka do najbližeg susjeda određuje se s Euklidovom udaljenosti prema (4-2):

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2} \tag{4-1}$$

Zbog svoje jednostavnosti, sama implementacija algoritma je vrlo jednostavna. Pseudokod k-NN algoritma prikazan je na slici 4.5

```
Učitaj trening skup i test skup
Odaberi vrijednost k
```

```
Za svaki podatak u testnom skupu:
        Pronađi Euklidovu udaljenost od svakog podataka iz trening skupa
        Spremi izračunate udaljenosti i sortiraj ih
        Odaberi prvih k udaljenosti
        Dodijeli klasu podatku iz test skupa prema većini najbližih udaljenosti trening
        skupa
```

Sl. 4.5 *Pseudokod k-NN algoritma*

4.5.3 Prikaz algoritma na Raspberry Pi-u

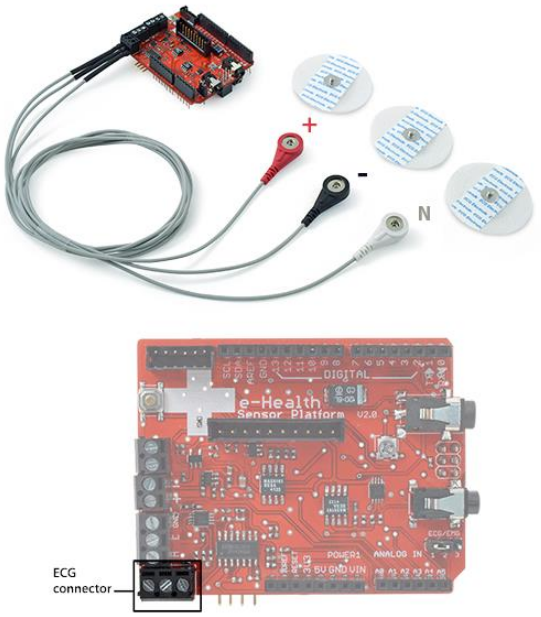
U ovom sustavu, k-NN algoritam pripadnost klasi određuje na temelju ulaznih podataka sa senzora, a na zaslon korisniku vraća klasu bolestan ili zdrav. Prikaz programskog koda za k-NN algoritam nalazi se na slici 4.6. Programski kod se izvršava na arhitekturi Raspberry Pi te se nalazi unutar glavnog programa koji vrši obradu podataka i izbor senzora za prikupljanje podataka.

```
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import numpy as np
    podaci=pd.read_csv('temp_mj.csv')
    X=podaci['temp']
    Y=podaci['status']
    x = np.array(X)
    y = np.array(Y)
    x = x.reshape(-1,1)
    classifier = KNeighborsClassifier(n_neighbors=2)
    classifier.fit(x,y)
```

Sl. 4.6 *Implementirani algoritam k-NN*

4.6 Snimanje EKG-a

Senzori za mjerenje EKG-a povezani su na *e-health* platformu pomoću EKG konektora koji se nalazi na samoj platformi. Senzori imaju polaritet pa je potrebno paziti na ispravne polarnosti prilikom spajanja (Sl. 4.6) [17].



Sl. 4.6 *Spajanje senzora za EKG na E-health platformu [17]*

Nakon povezivanja senzora i platforme na Arduino moguće je obaviti mjerenje. Pomoću biblioteke Ehealth.h i programskog koda (Sl. 4.7) omogućeno je dohvaćanje podataka sa senzora u obliku tablice gdje jedan stupac predstavlja vrijeme a drugi napon. Iz dobivenih vrijednosti se pravi graf koji je zapravo EKG to jest elektrokardiogram. Za dohvaćanje podataka sa senzora potrebno je koristiti eHealth.h biblioteku koju proizvođač Libelium pruža uz svoje *e-health* platforme te iz te biblioteke funkciju getECG, koja omogućuje prijenos podataka s EKG senzora na Arduino. Ostatak programskog koda prikazanog na slici 4.7 služi za slanje izmjerenih podataka s Arduina na Raspberry Pi.

```
#include <eHealth.h>                          //Libelium biblioteka potrebna za rad platforme
void ECGBucle(void) {
      int i;
      float c=0;
      for (i=0;i<1000;i++){
         float ECG = eHealth.getECG();   //Libelium funkcija za dohvaćanje podataka sa senz.
         Serial.print(c/67, 2);                //Slanje vremenskog trajanja
         Serial.print(" ");
         Serial.print(ECG, 2);                //Slanje EKG vrijednosti
         Serial.println("");
         c++;
         delay(15);
         }
      mainMenu();                             //Povratak u glavni izbornik
    }
```

Sl. 4.7 *Programski kod za dohvaćanje podataka sa EKG senzora*

Obrada dobivenih podataka vrši se na Raspberry Pi-u, na kojeg se podaci u stvarnom vremenu šalju te se primljeni podaci spremaju u dokument u obliku tablice s dva stupca. Prvi stupac predstavlja trajanje mjerenja, a drugi dobivenu naponsku vrijednost. Nakon završetka mjerenja iz dobivenih podataka sustav generira graf pomoću programskog koda (Sl. 4.8) te sprema generirani graf u obliku slike i prikazuje ga na zaslonu.

```
arudinoSerialData.write(b'2')                          #aktivacija senzora
time.sleep(0.01)
file=open("EKG.txt","w")                               #kreiranje datoteke
for i in range (1,1000):
      if(arudinoSerialData.inWaiting()>0):
         test = arudinoSerialData.readline()           #dohvacanje podataka sa Arduina
         print test
         file.write(test)
         time.sleep(0.015)
file.close()
with open('EKG.txt', 'r') as f:
   lines = f.readlines()
   x = [float(line.split()[0]) for line in lines]      #odabir stupca
   y = [float(line.split()[1]) for line in lines]
plt.plot(x ,y)                                         #generiranje grafa
plt.show()                                             #prikaz grafa
```

Sl. 4.8 *Programski kod za obradu podataka sa EKG senzora*

4.7 Mjerenje tjelesne temperature

Senzor za mjerenje temperature (Sl. 4.9) povezuje se na *e-health* platformu preko 3.5 milimetarskog pristupa na platformi. Zatim se senzor spaja na ljudsko tijelo, najčešće na kažiprst. Samom senzoru je potrebno neko vrijeme kako bi se mjerenja stabilizirala, zbog nagle promjene temperature. Obično je to vrijeme oko jedne minute. Mjerni podaci dobivaju se pomoću *e-health* biblioteke i programskog koda (Sl. 4.10) u obliku tablice gdje je jedan stupac vrijeme a drugi temperatura [17]. Za dohvaćanje podataka sa senzora potrebno je koristiti eHealth.h biblioteku koju proizvođač Libelium pruža uz svoje *e-health* platforme te iz te biblioteke funkciju getTemperature, koja omogućuje prijenos podataka sa senzora za mjerenje tjelesne temperature na Arduino. Ostatak programskog koda prikazanog na slici 4.10 služi za slanje izmjerenih podataka s Arduina na Raspberry Pi.

Sl. 4.9 *Senzor za mjerenje tjelesne temperature [17]*

```
#include <eHealth.h>                    //Libelium biblioteka potrebna za rad platforme
void temperatureBucle(void)
   {
      int i;
      for (i=0;i<6;i++){
         float temperature = eHealth.getTemperature();      //Libelium funkcija
         Serial.println(temperature, 2);                    //Slanje temperaturne vrijednosti
         delay(1000);
       }
      mainMenu();                                           //Povratak u glavni izbornik
}
```

Sl. 4.10 *Programski kod za dohvaćanje podataka sa temperaturnog senzora*

Dobiveni podaci sa senzora u stvarnom vremenu se prenose s Arduina na Raspberry Pi te se spremaju u tekstualnu datoteku i na spremljenim podacima se vrši k-NN metoda strojnog učenja. Nakon provedene metode strojnog učenja korisnika sustav obavještava o izmjerenoj temperaturi te o njegovom zdravstvenom stanju. Programski kod nalazi se na slici 4.11.

```
      arudinoSerialData.write(b'1')                          #aktivacija senzora
      podaci=pd.read_csv('temp_mj.csv')                      #odabir datoteke za strojno ucenje
      X=podaci['temp']                                       #odabir podataka
      Y=podaci['status']
      x = np.array(X)
      y = np.array(Y)
      x = x.reshape(-1,1)
      classifier = KNeighborsClassifier(n_neighbors=2)       #odabir broja susjeda
      classifier.fit(x,y)
      l=1
      while l==1:
         for i in range (1,6):
            if(arudinoSerialData.inWaiting()>0):
               test = arudinoSerialData.readline()           #dohvaćanje podataka sa Arduina
               print test
               a=classifier.predict([[test]])                #obrada podataka sa Arduina
               if a==0:
```

```
            print "Temperatura nije povisena\n"
        else:
            print "Temperatura je povisena"
        time.sleep(1)
l=0
```

Sl. 4.11 *Programski kod za obradu podataka sa temperaturnog senzora*

4.8 Mjerenje krvnog tlaka

Mjerenje krvnog tlaka obavlja se pomoću tlakomjera koji nije direktno spojen na e-*health* platformu, zbog ograničenja platforme a i samog načina rada tlakomjera. Potrebno je imati zaseban uređaj to jest tlakomjer kojim bi korisnik mogao obaviti mjerenja. Uređaj se sastoji od zaslona i manšete koja se napuhuje. Napuhivanje manšete omogućuje mjerenje sistoličkog i dijastoličkog tlaka pomoću oscilometrijske detekcije, koristeći ili deformabilne membrane koje se mjere pomoću diferencijalne kapacitivnosti ili diferencijalne otpornosti, te uključuju mikroprocesor [20]. Unatoč nedostacima *e-health* platforme, sustav i dalje može pratiti krvni tlak korisnika, no potreban je ručni unos dobivenih mjerenja u sustav. Sustav na zaslonu od korisnika traži unos izmjerenih podataka te nad njima radi obradu. Uneseni podaci se uspoređuju s unaprijed poznatim granicama ljudskog krvnog tlaka [3] te na osnovi istih vrši se obavještavanje korisnika. Slika 4.12 prikazuje programski kod potreban za to ostvarenje.

```
a=int(input("Upisite izmjerenu vrijednost sistolickog tlaka: "))
b=int(input("Upisite izmjerenu vrijednost dijastolickog tlaka: "))
if a>140 or b>90:
    print("Krvni tlak je povisen")
elif a<100 or b<60:
    print("Krvni tlak je snizen")
else:
    print("Krvni tlak je u redu")
if a>180 or b>110:
    print("Krvni tlak je opasno povišen, potražite medicinsku pomoć!")
```

Sl. 4.12 *Programski kod za obradu podataka krvnog tlaka*

4.9 Mjerenje elektrodermalne aktivnosti

Senzori za elektrodermalnu aktivnost povezuju se na platformu pomoću dva priključka (Sl. 4.13). Senzori nemaju polarizaciju tako da nije bitno koji senzor ide u koji priključak. Povezuju se na prste ljudskog tijela, puštaju struju kroz prste te mjere otpor i vodljivost ljudskog tijela. Dobiveni rezultati su numeričkog oblika te je iz njih moguće napraviti graf [17]. Za dohvaćanje

podataka sa senzora potrebno je koristiti eHealth.h biblioteku koju proizvođač Libelium pruža uz svoje *e-health* platforme te iz te biblioteke funkciju getSkinConductanceVoltage, koja omogućuje prijenos podataka sa senzora za mjerenje elektrodermalne aktivnosti na Arduino. Ostatak programskog koda prikazanog na slici 4.14 služi za slanje izmjerenih podataka s Arduina na Raspberry Pi.

Sl. 4.13 *Senzori za mjerenje elektrodermalne aktivnosti [17]*

```
#include <eHealth.h>                          //Libelium biblioteka potrebna za rad platforme
void skinSensorBucle(void) {
    int i;
    float c=0;
    for (i=0;i<20;i++){
       float conductanceVol = eHealth.getSkinConductanceVoltage();     //Libelium funkcija
       Serial.print(c/2, 2);                  //Slanje vremenskog trajanja
       Serial.print(" ");
       Serial.print(conductanceVol, 2);    //Slanje vrijednosti provodljivosti
       Serial.println("");
       c++;
       delay(500);
      }
    mainMenu();                    //Povratak u glavni izbornik
}
```

Sl. 4.14 *Programski kod za dohvaćanje podataka sa senzora za elektrodermalnu aktivnost*

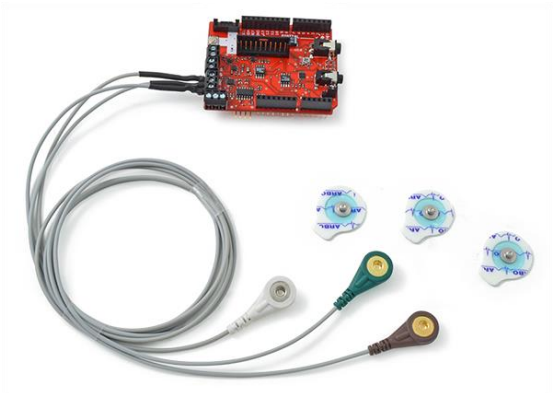
Mjerenja dobivena sa senzora su u stvarnom vremenu te se šalju na Raspberry Pi na kojemu se vrši obrada podataka. Podaci se spremaju u tekstualni dokument u obliku tablice s dva stupca. Prvi stupac predstavlja vrijeme mjerenja, to jest trajanje mjerenja, a drugi stupac predstavlja vodljivost kože izraženu u mikro simensima. Nakon završetka mjerenja iz dobivenih podataka sustav generira graf pomoću programskog koda (Sl. 4.15) te sprema generirani graf u obliku slike i prikazuje ga na zaslonu.

```
arudinoSerialData.write(b'3')                                   #aktivacija senzora
time.sleep(0.01)
file=open("GSR.txt","w")                                        #kreiranje datoteke
for i in range (1,20):
      if(arudinoSerialData.inWaiting()>0):
         test = arudinoSerialData.readline()          #dohvaćanje podataka sa Arduina
         print test
         file.write(test)
         time.sleep(0.5)
file.close()
with open('GSR.txt', 'r') as f:
   lines = f.readlines()
   x = [float(line.split()[0]) for line in lines]                #odabir stupca
   y = [float(line.split()[1]) for line in lines]
plt.plot(x ,y)                                                  #generiranje grafa
plt.show()                                                      #prikaz grafa
```

Sl. 4.15 *Programski kod za obradu podataka sa elektrodermalnog senzora*

4.10 Elektromiografija

Senzori za elektromiografiju spajaju se na isti način kao i sonde za EKG (Sl. 4.16). Zbog ograničenje platforme i načina rada, snimanje EKG-a i EMG-a nije moguće u isto vrijeme te je potrebno na platformi promijeniti postavke s EKG načina rada na EMG. Senzori se spajaju na biceps ljudskog tijela. Rezultati dobiveni su u obliku tablice gdje je jedan stupac vrijeme, a drugi stupac intenzitet struje koji prolazi kroz mišiće. Pretvaranjem podataka u graf dobijemo elektromiograf [17]. Za dohvaćanje podataka sa senzora potrebno je koristiti eHealth.h biblioteku koju proizvođač Libelium pruža uz svoje *e-health* platforme te iz te biblioteke funkciju getEMG, koja omogućuje prijenos podataka s EMG senzora na Arduino. Ostatak programskog koda prikazanog na slici 4.17 služi za slanje izmjerenih podataka s Arduina na Raspberry Pi.



Sl. 4.16 *Senzori za elektromiografiju [17]*

```
#include <eHealth.h>                    //Libelium biblioteka potrebna za rad platforme
void bodyPositionBucle(void){
    int i;
    float c=0;
    for (i=0;i<200;i++){
      int EMG = eHealth.getEMG();       //Libelium funkcija
      Serial.print(c/10, 2);            //Slanje vremenskog trajanja
      Serial.print(" ");
      Serial.print(EMG);                //Slanje EMG vrijednosti
      Serial.println("");
      c++;
      delay(100);
    }
    mainMenu();                         //Povratak u glavni izbornik
  }
```

Sl. 4.17 *Programski kod za dohvaćanje podataka sa EMG senzora*

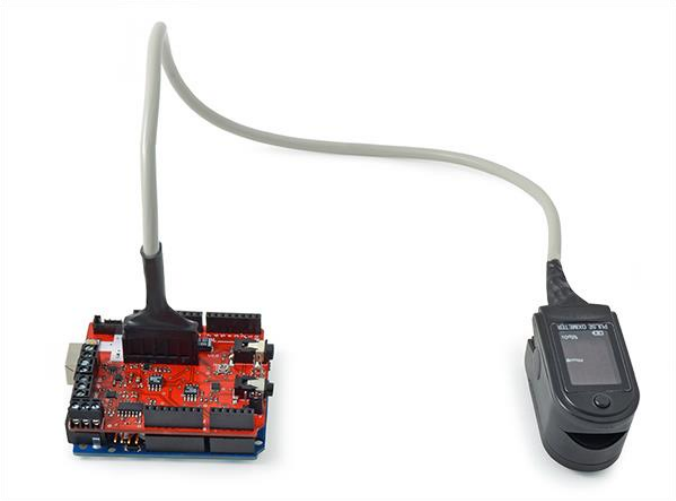
Dobivena mjerenja s EMG senzora su u stvarnom vremenu te se mjerenja s *e-health* platforme šalju na Raspberry Pi za obradu. Obrada podataka započinje spremanjem podataka u tekstualnu datoteku u obliku tablice s dva stupca. Prvi stupac tablice predstavlja vrijeme mjerenja, to jest trajanje mjerenja dok drugi stupac predstavlja naponsku vrijednost u mikro voltima. Nakon završetka mjerenja iz dobivenih podataka sustav generira graf pomoću programskog koda (Sl. 4.18) te sprema generirani graf u obliku slike i prikazuje ga na zaslonu.

```
    arudinoSerialData.write(b'4')                                   #aktivacija senzora
    time.sleep(0.01)
    file=open("EMG.txt","w")                                        #kreiranje datoteke
    for i in range (1,200):
        if(arudinoSerialData.inWaiting()>0):
            test = arudinoSerialData.readline()                 #dohvaćanje podataka sa Arduina
            print test
            file.write(test)
            time.sleep(0.1)
    file.close()
    with open('EMG.txt', 'r') as f:
        lines = f.readlines()
        x = [float(line.split()[0]) for line in lines]                  #odabir stupca
        y = [float(line.split()[1]) for line in lines]
    plt.plot(x ,y)                                                  #generiranje grafa
    plt.show()                                                      #prikaz grafa
```

Sl. 4.18 *Programski kod za obradu podataka sa EMG senzora*

4.11 Puls i zasićenost krvi kisikom

Senzor za mjerenje pulsa i zasićenosti krvi kisikom spaja se na *e-health* platformu na konektor za pulsni oksimetar(Sl. 4.19). Rad senzora zasniva se na dva zasebna izvora svjetlosti. Prvi izvor svjetlosti je valne duljine 600nm, dok drugi izvor ima valnu duljinu od 940nm. Svjetlost se djelomično apsorbira u hemoglobinu te je moguće odrediti zasićenost krvi kisikom računajući apsorpciju svjetlosti na dvjema valnim duljinama.. Deoksigenizirani hemoglobin apsorbira svjetlost valne duljine 600nm a oksigenizirani hemoglobin svjetlost valne duljine 940nm [7].



Sl. 4.19 *Pulsni oksimetar [17]*

Zbog ograničenja senzora koji je bio na raspolaganju, nije moguće izmjerene podatke poslati sa senzora na *e-health* platformu. Sustav i dalje može vršiti obradu izmjerenih podataka, ali je potrebno ručno unositi izmjerene podatke, koji su prikazani na zaslonu pulsnog oksimetra. Obrada podataka vrši se tako da korisnik ručno unosi podatke koji su prikazani na zaslonu pulsnog oksimetra, te se ti podaci spremaju u tablicu. Podaci o broju otkucaja srca u minuti korišteni su za k-NN metodu strojnog učenja. Pomoću programskog koda (Sl. 4.20) sustav javlja korisniku ako postoje odstupanja od korisnikovog normalnog zdravstvenog stanja, to jest obavještava korisnika ako njegov puls ili zasićenost kisikom nisu unutar njegovih standardnih granica.

```
podaci=pd.read_csv('otkucaji.csv')                    #datoteka za strojno ucenje
X=podaci['bpm']                                       #odabir podataka
Y=podaci['status']
```

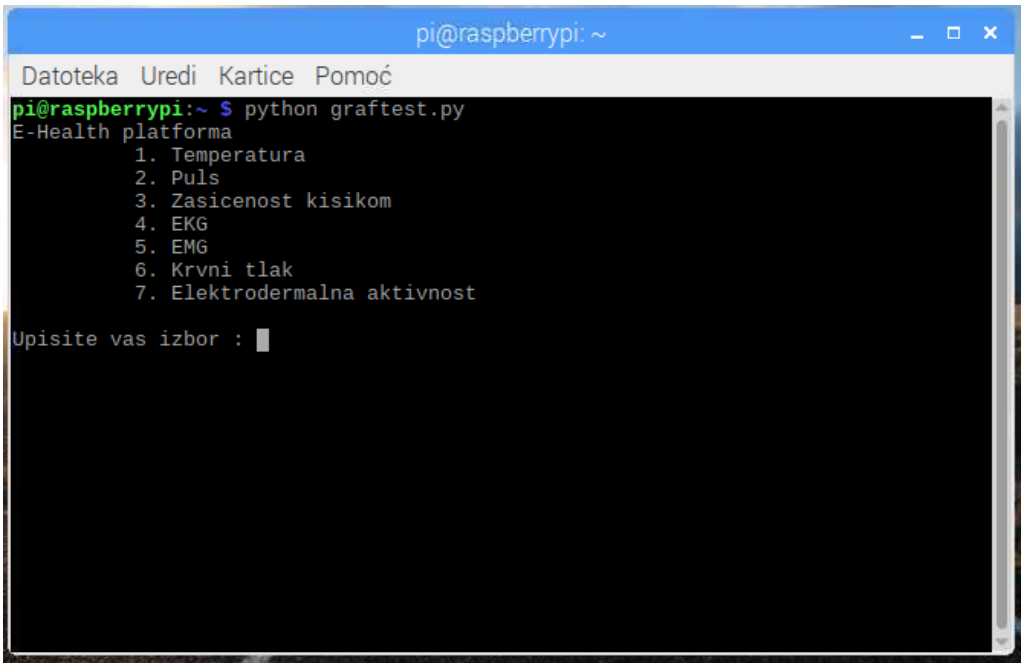
```
    x = np.array(X)
    y = np.array(Y)
    x = x.reshape(-1,1)
    classifier = KNeighborsClassifier(n_neighbors=2)              #odabir broja susjeda
    classifier.fit(x,y)
    test=float(input("Unesite broj otkucaja srca u minuti sa senzora: "))
    a=classifier.predict([[test]])                                #obrada podataka
    if a==0:
       print"\nOtkucaji srca su u normalnim granicama\n"
    else:
       print"\nOtkucaji srca odstupaju od normalnih granica\n"
 elif choice == 3:
    a=int(input("Upisite izmjerenu zasicenost krvi kisikom sa senzora: "))
    if a < b:
       print "\nZasicenost kisikom je snizena, moguci zdravstveni problemi\n"
    else:
       print "\nZasicenost kisikom je normalna\n"
```

Sl. 4.20 *Programski kod za obradu podataka sa pulsnog oksimetra*

Kroz poglavlje 4. objašnjena su sklopovska rješenja, te programska osnova potrebna za ostvarenje funkcionalnosti vlastitog sustava koji je detaljnije opisan u poglavlju 5. te je pušten u rad u stvarnim uvjetima.

5. PRIKAZ RADA SUSTAVA U STVARNIM UVJETIMA S REZULTATIMA I ANALIZOM

Nakon izrade sustava, sustav je stavljen u rad, to jest počelo je prikupljanje podataka kako bi se moglo provesti strojno učenje. Strojno učenje primijenjeno je na korisnikovu tjelesnu temperaturu tijela te puls. Za svaki od navedenih podataka provedeno je prikupljanje od 200 uzoraka. Za tih 200 uzoraka bilo je potrebno odrediti da li je korisnik bolestan ili ne, kako bi algoritam u budućnosti mogao sam odrediti zdravlje korisnika. Nakon toga sustav je pušten u rad s primijenjenim algoritmom za strojno učenje te je korisnik koristio sustav kroz nekoliko dana. Prilikom korištenja korisnik bira što želi mjeriti to jest koji biomedicinski podatak želi izmjeriti pomoću izbornika (Sl. 5.1). Zbog limitacija e-*health* platforme i Arduina nije moguće prikupljati više biomedicinskih podataka istovremeno [17].

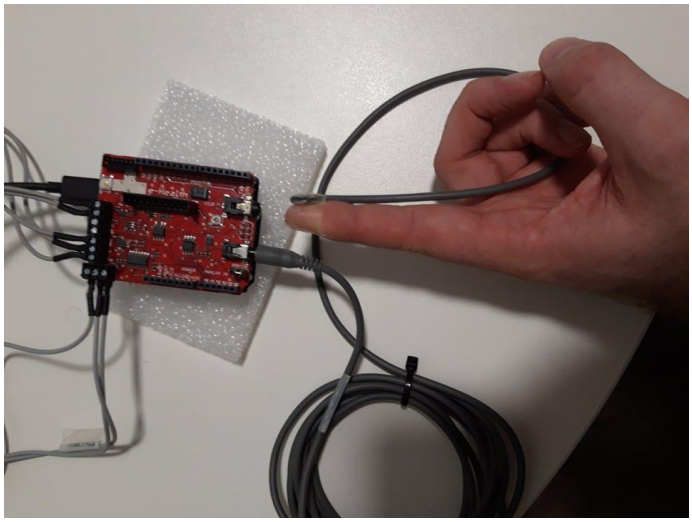


Sl. 5.1 *Glavni izbornik sustava*

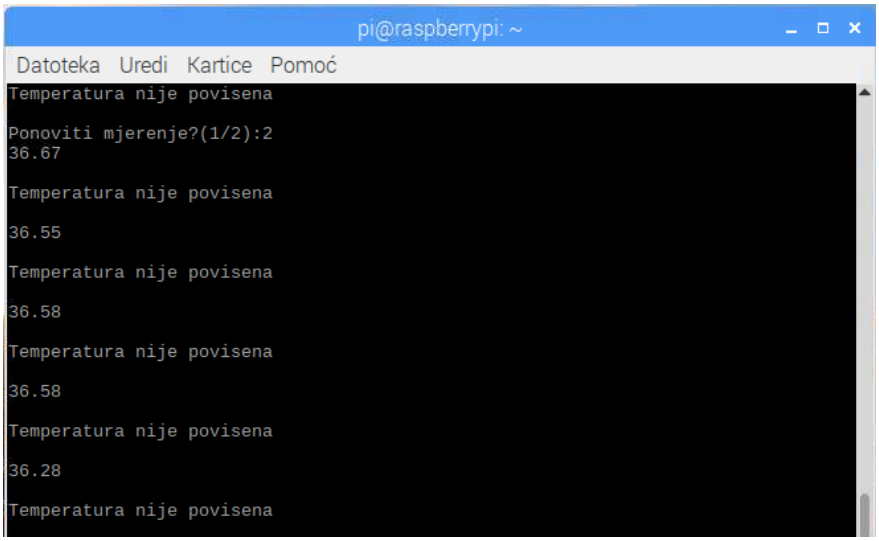
5.1 Mjerenje tjelesne temperature

Odabirom mjerenja tjelesne temperature, Arduino počinje prikupljati podatke sa senzora za mjerenje tjelesne temperature. Samom senzoru treba oko jedne minute kako bi počeo prikupljati točne podatke zbog razlike u temperaturi između ljudskog tijela i samog senzora. Senzor se spaja na korisnikov kažiprst te ga je potrebno ili držati sigurno na mjestu ili ga zalijepiti za prst (Sl. 5.2). Senzor šalje podatke na Arduino u brojčanom obliku koji predstavljaju temperaturu

ljudskog tijela u stupnjevima Celzijusa. Zatim se podaci prosljeđuju na Raspberry Pi gdje se ti podaci spremaju u tekstualnu datoteku. Nakon spremanja podataka, nad njima se vrši analiza u obliku strojnog učenja gdje sustav na temelju prošlih podataka određuje je li pacijent bolestan ili ne. Pacijent je obaviješten o svom stanju ispisom izmjerene temperature i zdravstvenim stanjem na zaslon. Moguće je ponoviti mjerenje odabirom opcije za ponavljanje mjerenja na zaslonu. Očekivane vrijednosti očitanja senzora su u rasponu od 34°C do 42°C.



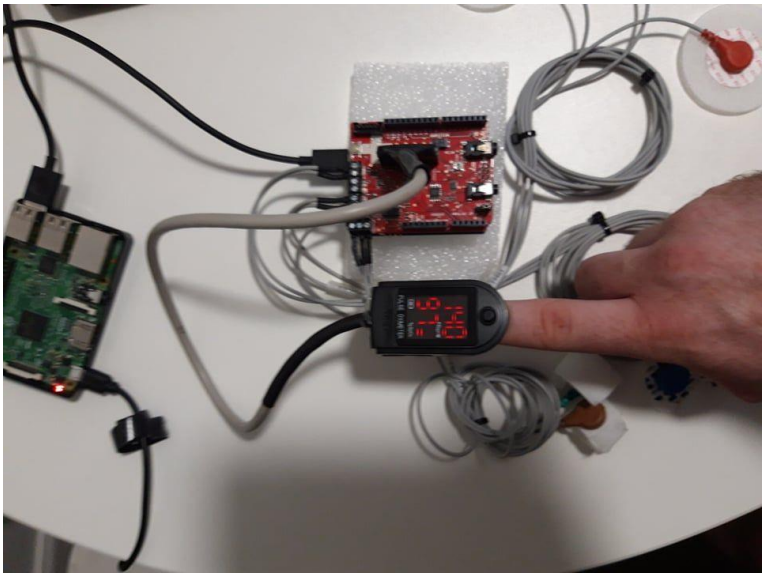
Sl. 5.2 *Spajanje senzora na korisnikov kažiprst*



Sl. 5.3 *Prikaz na zaslonu pri mjerenju temperature*

5.2 Mjerenje zasićenosti krvi kisikom i broja otkucaja srca u minuti

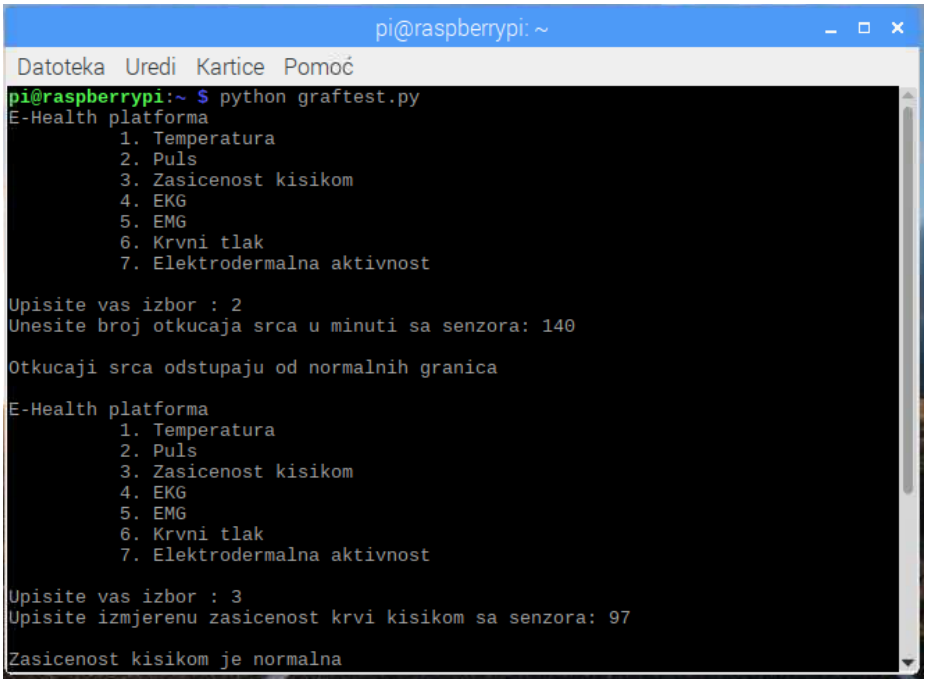
Prilikom prikupljanja podataka o zasićenosti krvlju kisikom te broju otkucaja srca u minuti potrebno je ručno upisati vrijednosti u program koji se izvodi na Raspberry-u zbog limitacija senzora, to jest senzor ne može slati podatke jer nema vodove za slati podatke nego samo *e-health* platformu koristi za napajanje senzora. Pulsni oksimetar se spaja na palac korisnika te se nakon spajanja pritisne tipka koja pali oksimetar i započinje mjerenje. Kada mjerenje završi na zaslonu oksimetra su ispisane izmjerene vrijednosti (Sl. 5.4). Senzor je u mogućnosti mjeriti podatke za zasićenost kisikom u rasponu od 0% do 100%, te broj otkucaja srca u minuti od 0 do 200. Očekivani rezultati sa senzora kreću se u rasponu od 90% do 100% za zasićenost krvi kisikom te od 50 do 150 za broj otkucaja srca u minuti.



Sl. 5.4 *Spajanje pulsnog oksimetra na korisnikov palac*

Na glavnom izborniku potrebno je odabrati puls za upis izmjerenih otkucaja srca nakon čega će sustav obavijestiti korisnika ako izmjereni podaci odstupaju od svakodnevnih korisnikovih podataka u zdravom stanju. Isto je potrebno učiniti i za zasićenost krvi kisikom što se postiže odabirom opcije zasićenost krvi kisikom u glavnom izborniku programa. Nakon ručnog upisa izmjerenog podatka o zasićenosti krvi kisikom sa senzora, sustav će upozoriti korisnika ako je zasićenost kisikom niska, odnosno ako zasićenost krvi kisikom padne ispod 90%, što se smatra stanjem hipoksije i može biti opasno po život [7]. Svi izmjereni podaci spremaju se u tekstualnu datoteku. Nakon spremanja podataka, nad podacima o broju otkucaja srca u minuti vrši se analiza metodom k-NN strojnog učenja. U slučaju zasićenosti kisikom nema potrebe za strojnim

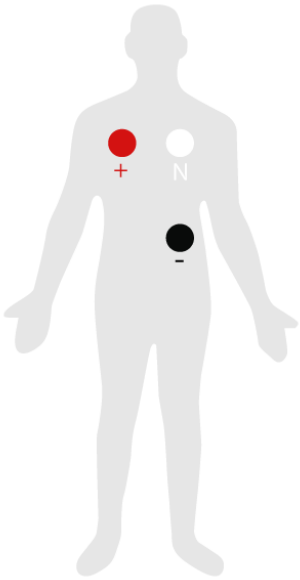
učenjem. Sustav na zaslon ispisuje zdravstveno stanje pacijenta te ga upozorava ako ijedan podatak odstupa od normalnih granica.



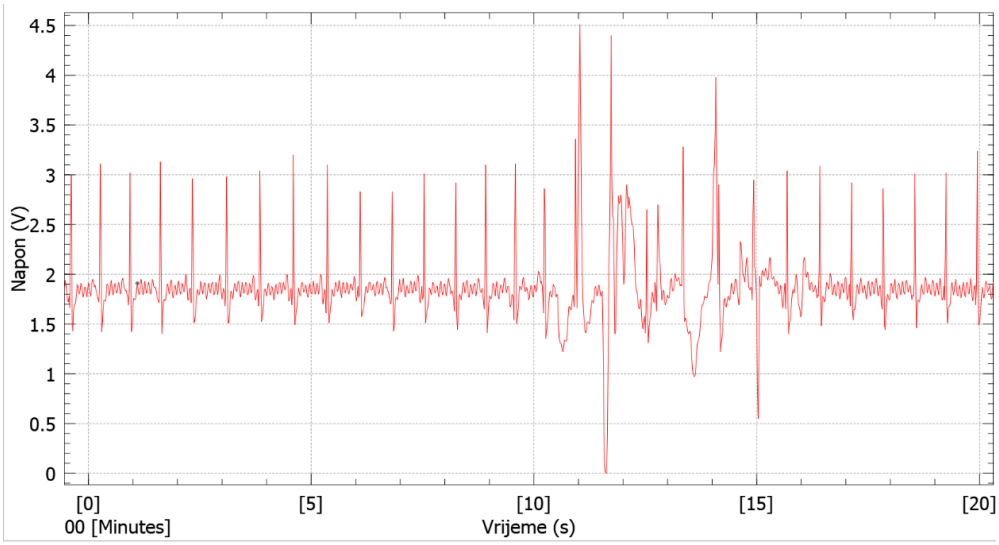
Sl. 5.3 *Prikaz na zaslonu pri mjerenju pulsa i zasićenosti kisikom*

5.3 Izrada EKG-a

Izrada elektrokardiograma korisnika moguća je odabirom EKG opcije u glavnom izborniku. Prije pokretanja EKG opcije iz glavnog izbornika prvo je potrebno spojiti senzore na korisnika (Sl. 5.4). Nakon što je svaki senzor pravilno spojen na korisnika, pokrenuti EKG opciju. U ovom programu prikupljanje podataka sa senzora za izradu EKG-a traje 20 sekundi. Moguće je produžiti trajanje ako to korisnik želi. Senzori na Arduino šalju podatke u brojčanom obliku gdje prvi broj predstavlja vrijeme mjerenja a drugi broj predstavlja izmjereni napon u mili voltima. Za vrijeme trajanja mjerenja podaci se prosljeđuju s Arduina na Raspberry Pi gdje se podaci spremaju. Podaci se spremaju u tekstualnu datoteku u obliku tablice s dva stupca od koji je prvi stupac vrijeme, odnosno trajanje mjerenja a drugi stupac napon u mili voltima. Iz spremljenih podataka sustav generira graf te ga sprema u memoriju i korisniku ga prikazuje na zaslonu. Dobivene vrijednosti sa senzora su u rasponu od 1mV do 5mV.



Sl. 5.4 *Spajanje EKG senzora na korisnika [17]*



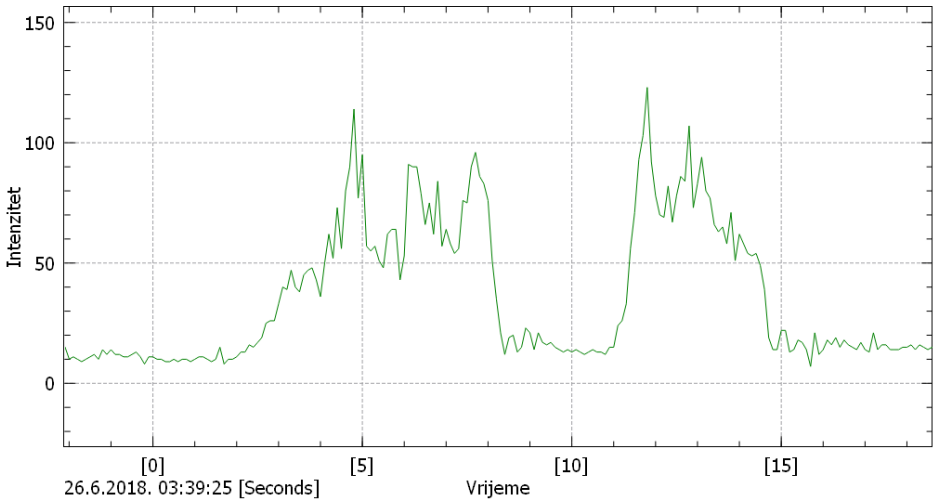
Sl. 5.5 *Izmjereni elektrokardiogram*

5.4 Izrada EMG-a

Za izradu elektromiografa potrebno je senzore prvo spojiti na korisnikov biceps (Sl. 5.6) te nakon toga na e-*health* platformi promijeniti opciju mjerenja iz EKG u EMG. Zatim treba na glavnom izborniku odabrati opciju EMG te pokrenuti program. Prikupljanje podataka sa senzora traje 20 sekundi. Moguće je produžiti trajanje ako to korisnik želi. Senzori na Arduino šalju podatke u brojčanom obliku gdje prvi broj predstavlja vrijeme mjerenja a drugi broj predstavlja izmjereni napon u mikro voltima. Za vrijeme trajanja mjerenja podaci se prosljeđuju s Arduina na Raspberry Pi gdje se podaci spremaju. Podaci se spremaju u tekstualnu datoteku u obliku

tablice s dva stupca od koji je prvi stupac vrijeme, odnosno trajanje mjerenja a drugi stupac napon u mikro voltima. Iz spremljenih podataka sustav generira graf te ga sprema u memoriju i korisniku ga prikazuje na zaslonu. Dobivene vrijednosti sa senzora su u rasponu od 30µV do 30mV.

Sl. 5.6 *Spajanje EMG senzora na korisnikov biceps [17]*

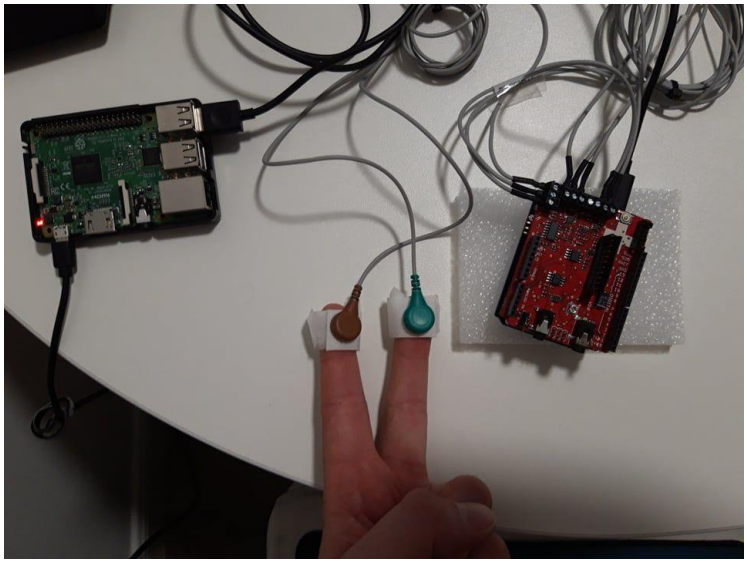


Sl. 5.7 *Izmjereni elektromiograf*

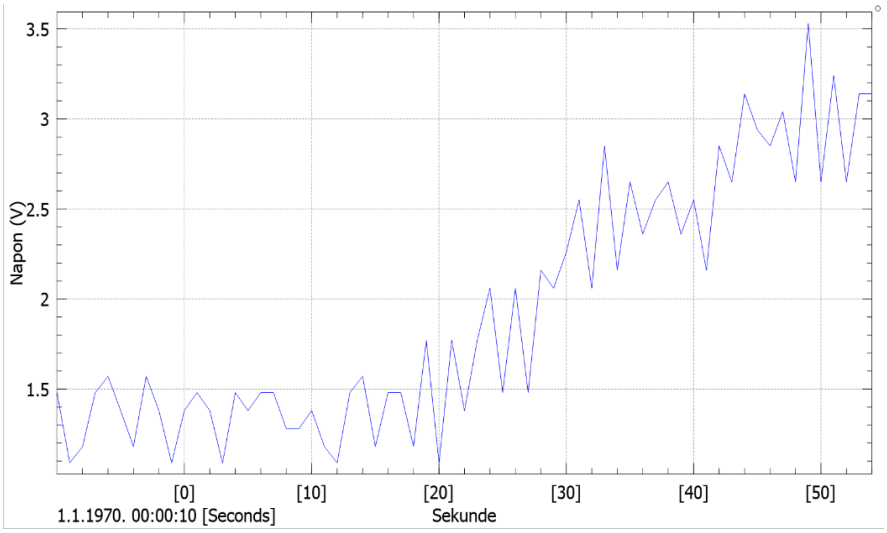
5.5 Izrada grafa elektrodermalne aktivnosti

Za izradu grafa elektrodermalne aktivnosti, kao i u prethodnim slučajevima potrebno je prvo spojiti mjerne senzore na korisnikov kažiprst i srednji prst (Sl. 5.8), te nakon toga odabrati opciju elektrodermalna aktivnost u glavnom izborniku. Prikupljanje podataka traje 20 sekundi no moguće je produžiti trajanje ako to korisnik želi. Senzori na Arduino šalju podatke u brojčanom obliku gdje prvi broj predstavlja vrijeme mjerenja a drugi broj predstavlja izmjereni napon u

mikro simensima. Za vrijeme trajanja mjerenja podaci se prosljeđuju s Arduina na Raspberry Pi gdje se podaci spremaju. Podaci se spremaju u tekstualnu datoteku u obliku tablice s dva stupca od koji je prvi stupac vrijeme, odnosno trajanje mjerenja a drugi stupac napon u mikro simensima. Iz spremljenih podataka sustav generira graf te ga sprema u memoriju i korisniku ga prikazuje na zaslonu. Dobivene vrijednosti sa senzora su u rasponu od 1µS do 4µS.



Sl. 5.8 *Spajanje senzora na korisnikov kažiprst i srednji prst*

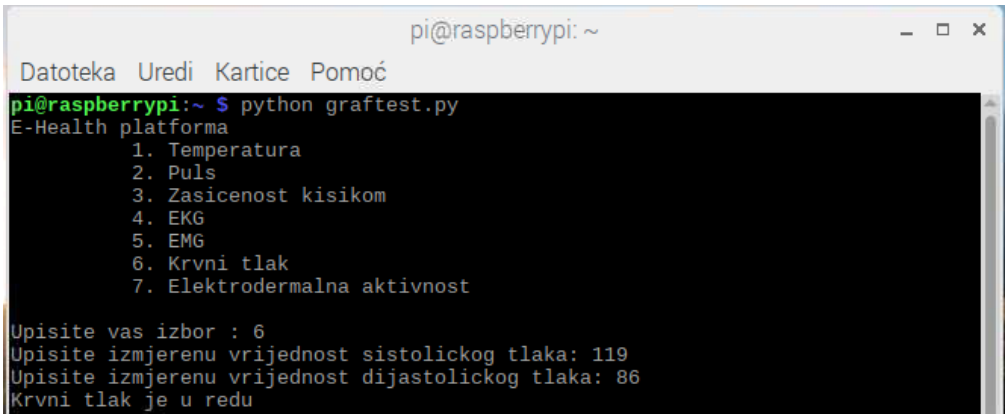


Sl. 5.9 *Izmjereni graf elektrodermalne aktivnosti*

5.6 Mjerenje krvnog tlaka

Mjerenje tlaka provodi se na tlakomjeru Kodea KD-202F koji radi zasebno bez obzira na *e-health* platformu [21]. Korisnik prvo izmjeri tlak koristeći samo tlakomjer te nakon završenog mjerenja u glavnom izborniku programa odabere opciju krvni tlak. Nakon odabira opcije korisnik sam unosi izmjerene vrijednosti prikazane na zaslonu tlakomjera (Sl. 5.10). Prilikom upisa korisnik mora unijeti dvije vrijednosti, sistoličku vrijednost tlaka i dijastoličku. Sistolička vrijednost je prva vrijednost na zaslonu tlakomjera a dijastolička druga. Nakon upisa vrijednosti sustav će spremiti upisane vrijednosti u tekstualnu datoteku na Raspberry Pi-u te provesti analizu. Analiza se sastoji od jednostavne usporedbe izmjerenih podataka s graničnim vrijednostima krvnog tlaka preuzetih iz medicinskog priručnika Hrvatskog liječničkog zbora. Očekivane vrijednosti s tlakomjera su u rasponu od 80 do 200 za sistolički tlak te 50 do 120 za dijastolički tlak. Granice normalnog funkcioniranja ljudskog tijela definirane u medicinskom priručniku su u rasponu od 100 do 140 za sistolički tlak te od 60 do 90 za dijastolički tlak Ako je detektirano odstupanje od normalnih vrijednosti, pacijent se upozorava ispisom poruke na zaslonu.

Sl. 5.10 *Izmjerene vrijednosti na zaslonu tlakomjera*



Sl. 5.11 *Prikaz na zaslonu nakon upisa izmjerenih vrijednosti krvnog tlaka*

5.7 Svrha strojnog učenja

Primjena strojnog učenja u ovom radu imala je funkciju boljeg prilagođavanja sustava korisniku. Kod nekih osoba, njihovi normalni biomedicinski podaci, poput broja otkucaja srca u minuti te tjelesne temperature, znaju odstupati od medicinski definiranih granica zdrave osobe. Tjelesna temperatura zdrave osobe specifična je za svaku osobu, te zbog toga postoje osobe kojima je normalna temperatura tijela oko 35°C što bi drugi sustavi mogli detektirati kao nisku temperaturu te upozoriti korisnika na mogućnost bolesti iako je to korisnikova normalna temperatura. Najizraženije razlike u broju otkucaja srca u minuti imaju sportaši. Profesionalni sportaši koji se bave sportovima poput plivanja, veslanja i bicikliranja imaju nizak broj otkucaja srca u minuti, najčešće oko 30 do 40. Zbog toga postojeća rješenja često obavijeste pacijenta da je bolestan no zapravo nije. Taj problem u ovom radu je riješen strojnim učenjem. Korisnik na temelju svojih mjerenja izradi trening skup, na kojem će se k-NN učiti. Tako može sam sebi prilagoditi sustav čime se postiže veća točnost i mogućnost prepoznavanja bolesti kod osoba čiji biomedicinski podaci pri zdravom stanju odstupaju od medicinski normalnih granica.

5.8 Prikaz dobivenih rezultata i način spremanja

Sustav je uspješno realiziran i dobivena su mjerenja sa svih korištenih senzora. Mjerenja su spremljena u odgovarajuće datoteke, to jest tjelesna temperatura je spremljena temp.csv, puls u otkucaji.csv, zasićenost krvi kisikom u o2.txt, podaci potrebni za izradu EKG-a u EKG.txt, EMG-a u EMG.txt, grafa elektrodermalne aktivnosti u GSR.txt te mjerenja krvnog tlaka u tlak.txt. Svaka od datoteka sastoji se od dva stupca, no sadržaji stupaca se različiti ovisno o tipu podataka koji se prikuplja. Prvih 10 mjerenja te spremljeni tipovi podataka i značenja stupaca

prikazani su u tablici 5.12. Stanje 0 u tablici označava da je korisnik zdrav, a stanje 1 da postoji mogućnost bolesti.

Tablica 5.12 Prikaz izmjerenih podataka

temp.csv		otkucaji.csv		o2.txt		tlak.txt	
°C	stanje	BPM	stanje	%	stanje	Sistola	Dijastola
36.50	0	92	0	99	0	129	89
36.20	0	81	0	97	0	118	80
37.20	0	130	1	100	0	119	86
38.00	1	101	0	98	0	124	84
38.10	1	93	0	99	0	107	75
35.90	0	86	0	98	0	116	79
35.80	0	129	1	97	0	135	92
37.40	0	140	1	99	0	112	74
36.50	0	103	0	96	0	140	95
36.90	0	84	0	100	0	132	90

Tablica 5.13 prikazuje kako su spremljena mjerenja potrebna za generiranje grafova.

Tablica 5.13 Spremljeni podaci za generiranje grafova

EKG.txt		EMG.txt		GSR.txt	
Vrijeme[s]	mV	Vrijeme[s]	µV	Vrijeme[s]	µS
0.00	1.88	0.00	15	0.00	1.48
0.02	1.94	0.10	10	0.10	1.09
0.04	1.88	0.20	11	0.20	1.18
0.06	1.80	0.30	10	0.30	1.48
0.08	1.73	0.40	9	0.40	1.57
0.10	1.72	0.50	10	0.50	1.38
0.12	1.78	0.60	11	0.60	1.18
0.14	1.66	0.70	12	0.70	1.57
0.16	2.48	0.80	10	0.80	1.38
0.18	3.00	0.90	14	0.90	1.09

5.9 Analiza rada sustava u stvarnim uvjetima

Za provjeru točnosti podataka sa senzora, izmjereni podaci uspoređeni su s podacima dobivenih s već postojećih rješenja za mjerenje biomedicinskih podataka. Dobivena mjerenja sa senzora za mjerenja tjelesne temperature uspoređena su s mjerenjima dobivenih s digitalnim toplomjerom. U rasponu od 36°C do 39°C odstupanja senzora od toplomjera bila su oko 0.5°C. Pri višim temperaturama od 39°C te nižima od 36°C pojavljuje se veće odstupanje koje je bilo u rasponu od 0.5°C do 0.8°C. EKG, EMG, elektrodermalnu aktivnost i zasićenost krvi kisikom nije bilo moguće usporediti s drugim uređajima. Broj otkucaja srca u minuti sa senzora uspoređen je sa

stvarnim brojem otkucaja srca u minuti, koji je dobiven vlastitim brojanjem otkucaja srca, te su odstupanja bila u rasponu od jednog do tri otkucaja razlike, što je gotovo zanemarivo odstupanje. Krvni tlak je mjeren koristeći medicinski certificiran uređaj tako da su njegove vrijednosti prihvaćene takvima kao točne bez odstupanja. Strojno učenje je također uspješno primijenjeno te je model dobro naučio raspoznavati zdravstveno stanje pacijenta. S obzirom na odstupanja temperaturnog senzora od stvarnih vrijednosti, k-NN se pokazala kao dobra ideja jer je model naučio prepoznati zdravstveno stanje pacijenta bez obzira na odstupanja temperaturnog senzora.

6. ZAKLJUČAK

Sustav za prikupljanje biomedicinskih podataka uspješno je ostvaren koristeći računalo Raspberry Pi mikroupravljač Arduino sa senzorskom platformom *e-health*. Izmjereni su i prikupljeni su biomedicinski podaci korisnika, poput tjelesne temperature, broja otkucaja srca u minuti, krvnog tlaka, zasićenosti krvi kisikom, izrada EKG-a, EMG-a i grafa elektrodermalne aktivnosti te je sustav obavijestio korisnika o njegovom zdravstvenom stanju. Programski dio sustava sastoji se od programa na Arduino mikroupravljaču koji služi da prikupljanje podataka sa senzora te slanje na Raspberry Pi računalo, te programa na Raspberry Pi-u koji služi za primanje podataka sa Arduina te spremanje i obradu tih podataka. Postupak strojnog učenja k-NN također je uspješno implementiran te je sustav u stanju prilagoditi se pacijentu i otkriti odstupaju li pacijentova tjelesna temperatura i puls od očekivanih standardnih vrijednosti. Glavne pogodnosti ovakvog sustava su njegova jednostavnost korištenja i mogućnost mjerenja više biomedicinskih podataka pomoću jednog uređaja. Također, primjena strojnog učenja unaprjeđuje rad sustava, a tim više, jer jeftini dostupni uređaji na tržištu ne koriste metode strojnog učenja za prilagođavanje korisniku. Uređaj je idealan za korisnike koji žele provjeriti neke pokazatelje zdravstvenog stanja u nekoliko koraka uređajem prihvatljive cijene, ali necertificirani uređaj poput ovoga ni na koji način ne može zamijeniti profesionalne medicinske uređaje i ulogu liječnika. Ispitivanjem rada uređaja u stvarnim uvjetima utvrđeno je da sustav uspješno prikuplja podatke sa korištenih senzora, analizira ih te obavještava korisnika o njegovom zdravstvenom stanju.

Glavni problem kod korištenja razvijenog sustava je da nije moguće mjerenje podataka s više senzora istovremeno, odnosno samo jedan od senzora može mjeriti podatke. Ako pacijent želi mjeriti neki drugi biomedicinski podatak potrebno je vratiti se natrag u glavni izbornik te odabrati željeni senzor. Taj problem nije bilo moguće riješiti u sklopu izrade ovog rada zbog korištenog mikroupravljača i senzorske platforme. Korištenjem dodatnog mikroupravljača i senzorske platforme ovaj problem bi se mogao riješiti. Drugi problem koji se javlja je točnost senzora i spomenuta činjenica da korišteni senzori nemaju medicinske certifikate i nisu prošli nikakva medicinska testiranja. Odstupanja od stvarnih vrijednosti variraju između 5% i 10%. zato je razvijeni sustav moguće koristiti samo u informativne svrhe.

7. LITERATURA

[1] Medicinska dijagnostika, http://web.studenti.math.pmf.unizg.hr/~sakarme/meddijagnostika.html, datum pristupa 30.6.2019

[2] K. Aswini, ECG in 100 steps!, http://www.lifehugger.com/doc/120/ecg-100-steps, datum prisupa 30.6.2019

[3] Vrućica, akutna, u odraslih, http://www.msd-prirucnici.placebo.hr/msd-simptomi/vrucica-akutna-u-odraslih, datum pristupa 30.6.2019

[4] Visoki krvni tlak, http://www.msd-prirucnici.placebo.hr/msd-za-pacijente/bolesti-srca-i-krvnih-zila/visoki-krvni-tlak, datum pristupa 30.6.2019

[5] W. Boucsein, Electrodermal Activity, Springer Science & Business Media, Wuppertal, 2012., dostupno na https://books.google.hr/books/about/Electrodermal_Activity.html?id=6N6rnOEZEEoC, datum pristupa 30.6.2019

[6] Electromyography, https://en.wikipedia.org/wiki/Electromyography, datum pristupa 30.6.2019

[7] Pulsna oksimetrija, http://www.msd-prirucnici.placebo.hr/msd-prirucnik/pulmologija/testovi-plucne-funkcije/pulsna-oksimetrija, datum pristupa 30.6.2019

[8] E. Jovanov, Stress Monitoring Using a Distributed Wireless Intelligent Sensor System, IEEE Engineering in Medicine and Biology Magazine, Issue 3, Vol. 22, 49 - 55, May 2003, dostupno na https://ieeexplore.ieee.org/document/1213626, datum pristupa 30.6.2019

[9] Monitoring on the Move, https://philipsproductcontent.blob.core.windows.net/assets/20170609/ec37a17c4e4240848d70a78d0100e0ca.pdf, datum pristupa 30.6.2019

[10] CARESCAPE Monitor B850, https://www.gehealthcare.com/products/patient-monitoring/patient-monitors/carescape-monitor-b850, datum pristupa 30.6.2019

[11] Monitoring on the Go, https://philipsproductcontent.blob.core.windows.net/assets/20170523/b349908a9e374f02a157a77c016a06b7.pdf, datum pristupa 30.6.2019

[12] Surveyor™ S4 Mobile Monitor ,

https://www.welchallyn.com/en/products/categories/patient-monitoring/telemetry/surveyor-s4-wearable-monitor.html, datum pristupa 30.6.2019

[13] S. Sidhu, J. E. Marine, Evaluating and Managing Bradycardia, Trends in Cardiovascular Medicine, Issue 5, Volume 29, 249–312, July 2019, dostupno na https://www.sciencedirect.com/science/article/pii/S1050173819300933, datum pristupa 4.9.2019

[14] A. Umesh Mahtani, D. Gopinath Nair, Supraventricular Tachycardia, Medical Clinics of North America, Issue 5, Volume 103 863-879, September 2019, dostupno na https://www.sciencedirect.com/science/article/abs/pii/S0025712519300550 [4.9.2019]

[15] S, McManus, Raspberry Pi For Dummies, John Wiley & Sons,Hoboken, NJ, 2013

[16] M, Margolis, Arduino Cookbook, O'Reilly Media, Sebastopol, 2011

[17] e-Health Sensor Platform V2.0 for Arduino and Raspberry Pi [Biometric / Medical Applications], https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor- platform-arduino-raspberry-pi-medical, datum pristupa 30.6.2019

[18] Machine Learning, https://en.wikipedia.org/wiki/Machine_learning, datum pristupa 30.6.2019

[19] N. S. Altman, An Introduction to Kernel and Nearest-neighbor Nonparametric Regression, The American Statistician, Ithaca, 1992

[20] K-nearest Neighbors, http://www.statsoft.com/textbook/k-nearest-neighbors, datum pristpa 30.6.2019

[21] KD-202F, https://zowacare.com/product/kd-202f/, datum pristupa 30.6.2019

[22] T. Szydło, M. Konieczny, Mobile and Wearable Devices in an Open and Universal System for Remote Patient Monitoring, Microprocessors and Microsystems, Vol. 46, 44-56, October 2016, dostupno na https://www.sciencedirect.com/science/article/abs/pii/S0141933116300898?via%3Dihub, datum pristupa 4.9.2019

[23] J. A. Castellanos-Garzónac, E. Costac José, L. Jaimes, J. M.Corchadoa, An Evolutionary Framework for Machine Learning Applied to Medical Data, Knowledge-Based Systems, Volume 178, 111-122, August 2019, dostupno na https://www.sciencedirect.com/science/article/abs/pii/S0950705119304046, datum pristupa 4.9.2019

SAŽETAK

U današnje vrijeme, medicina se zasniva na velikom broju različitih biomedicinskih podataka koji su preduvjet uspješnog dijagnosticiranja i liječenja. Prikupljanje i analiza tih podataka ključni su za medicinu, te je mali dio tog problema prikazan i u ovom radu. Razvijeni sustav omogućuje prikupljanje podataka poput tjelesne temperature, krvnog tlaka, zasićenosti krvi kisikom, pulsa, izrada EKG-a, izrada EMG-a te izrada grafa elektrodermalne aktivnosti kože. Sustav je zasnovan na mikroupravljaču Arduino s pripadajućom senzorskom platformom, a on komunicira s računalom Raspberry Pi na kojem se obavlja obrada podataka. Programski dio sustava sastoji se od dva programa, jedan za prikupljanje podataka sa senzora, a drugi za analizu prikupljenih podataka. Postupak strojnog učenja k-NN primijenjen je na tjelesnu temperaturu te puls, odnosno da bi uređaj mogao pratiti korisnikove podatke kroz određen vremenski period i prilagoditi se korisniku. Uređaj također obavješćuje korisnika ukoliko bilo koji od izmjerenih podataka odstupa od normalnih vrijednosti. Ispitivanjem rada uređaja u stvarnim uvjetima utvrđeno je da uređaj ima odstupanja od stvarnih vrijednosti oko 5% te da se sustav može koristiti samo u informativne svrhe.

Ključne riječi: Arduino**,** biomedicinski podaci, prikupljanje podataka, Raspberry Pi, strojno učenje.

ABSTRACT

Development of data acquiring, monitoring and analyzing system for patient health monitoring

Nowadays, medicine is based on a large number of different biomedical data that are a prerequisite for successful diagnosis and treatment. Acquiring the biomedical data and analyzing it is crucial for medicine, and a small part of this problem is presented in this paper. The developed system enables the collection of data such as body temperature, blood pressure, oxygen saturation, pulse, ECG, EMG and skin electrodermal activity graph. The system is based on an Arduino microcontroller with an associated sensor platform, and it communicates with Raspberry Pi computer on which data is processed. Software part of the system is comprised of two programs, one for collecting data from sensors and the other for analyzing collected data. The k-NN machine learning algorithm was applied to body temperature and pulse, that is, the device could track the user's data over a period of time and adapt to the user. The device also notifies the user if any of the measured data deviates from normal values. When testing the device in real conditions, it was found that the device has deviations from the actual values of about 5% and that the system can be used for informational purposes only.

Keywords: Arduino**,** biomedical data, data collection, Raspberry Pi, machine learning

ŽIVOTOPIS

Bruno Brekalo rođen je 2.2.1994 u Zagrebu. U rujnu 2008. godine upisuje Gimnaziju Vukovar koju završava 2012. godine. Iste godine upisuje sveučilišni preddiplomski studij računarstva na Elektrotehničkom fakultetu u Osijeku. Završetkom preddiplomskog studija 2016. godine stječe zvanje: sveučilišni prvostupnik (baccalaureus)inženjer računarstva. Obrazovanje nastavlja iste godine upisom sveučilišnog diplomskog studija računarstva, smjer Procesno računarstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

Bruno Brekalo

PRILOZI

Prilog 1. Programski kod programa na Raspberry Pi-u

```
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import numpy as np
import serial
import matplotlib.pyplot as plt
arudinoSerialData = serial.Serial('/dev/ttyACM0',9600)
import time
b=95
while True:
   print("""E-Health platforma
        1. Temperatura
        2. Puls
        3. Zasicenost kisikom
        4. EKG
        5. EMG
        6. Krvni tlak
        7. Elektrodermalna aktivnost
        """)
   choice = raw_input('Upisite vas izbor : ')
   choice = int(choice)

   if choice == 1:
      arudinoSerialData.write(b'1')
      podaci=pd.read_csv('temp_mj.csv')
      X=podaci['temp']
      Y=podaci['status']
      x = np.array(X)
      y = np.array(Y)
      x = x.reshape(-1,1)
      #y = y.reshape(-1,1)
      #print x
      #print y
      classifier = KNeighborsClassifier(n_neighbors=2)
      classifier.fit(x,y)
      #print classifier.predict(35.41)
      l=1
      while l==1:
         for i in range (1,6):
            if(arudinoSerialData.inWaiting()>0):
               test = arudinoSerialData.readline()
               print test
               a=classifier.predict([[test]])
               if a==0:
                  print "Temperatura nije povisena\n"
```

```
            else:
                print "Temperatura je povisena"
            time.sleep(1)
        l=0

elif choice == 2:
    podaci=pd.read_csv('otkucaji.csv')
    X=podaci['bpm']
    Y=podaci['status']
    x = np.array(X)
    y = np.array(Y)
    x = x.reshape(-1,1)
    #y = y.reshape(-1,1)
    #print x
    #print y
    classifier = KNeighborsClassifier(n_neighbors=2)
    classifier.fit(x,y)
    test=float(input("Unesite broj otkucaja srca u minuti sa senzora: "))
    a=classifier.predict([[test]])
    if a==0:
        print"\nOtkucaji srca su u normalnim granicama\n"
    else:
        print"\nOtkucaji srca odstupaju od normalnih granica\n"

elif choice == 3:
    a=int(input("Upisite izmjerenu zasicenost krvi kisikom sa senzora: "))
    if a < b:
        print "\nZasicenost kisikom je snizena, moguci zdravstveni problemi\n"
    else:
        print "\nZasicenost kisikom je normalna\n"

elif choice==4:
    arudinoSerialData.write(b'2')
    time.sleep(0.01)
    file=open("EKG.txt","w")
    for i in range (1,1000):
          if(arudinoSerialData.inWaiting()>0):
              test = arudinoSerialData.readline()
              print test
              file.write(test)
              time.sleep(0.015)
    file.close()
    with open('EKG2.txt', 'r') as f:
        lines = f.readlines()
        x = [float(line.split()[0]) for line in lines]
        y = [float(line.split()[1]) for line in lines]
    plt.plot(x ,y)
    plt.show()

elif choice==5:
```

```
    arudinoSerialData.write(b'4')
    time.sleep(0.01)
    file=open("EMG.txt","w")
    for i in range (1,200):
          if(arudinoSerialData.inWaiting()>0):
             test = arudinoSerialData.readline()
             print test
             file.write(test)
             time.sleep(0.1)
    file.close()
    with open('EMG.txt', 'r') as f:
       lines = f.readlines()
       x = [float(line.split()[0]) for line in lines]
       y = [float(line.split()[1]) for line in lines]
    plt.plot(x ,y)
    plt.show()

elif choice==6:
    a=int(input("Upisite izmjerenu vrijednost sistolickog tlaka: "))
    b=int(input("Upisite izmjerenu vrijednost dijastolickog tlaka: "))
    if a>125 or b>85:
       print("Krvni tlak je povisen")
    elif a<100 or b<60:
       print("Krvni tlak je snizen")
    else:
       print("Krvni tlak je u redu")
                if a>180 or b>110:
                        print("Krvni tlak je opasno povišen, potražite medicinsku pomoć!")

elif choice==7:
    arudinoSerialData.write(b'3')
    time.sleep(0.01)
    file=open("GSR.txt","w")
    for i in range (1,20):
          if(arudinoSerialData.inWaiting()>0):
             test = arudinoSerialData.readline()
             print test
             file.write(test)
             time.sleep(0.5)
    file.close()
    with open('GSR.txt', 'r') as f:
       lines = f.readlines()
       x = [float(line.split()[0]) for line in lines]
       y = [float(line.split()[1]) for line in lines]
    plt.plot(x ,y)
    plt.show()
```

Prilog 2. Programski kod programa na Arduinu

```
#include <eHealth.h>
#include <PinChangeInt.h>
char serialByte;
uint8_t state = 0;
uint8_t cont = 0;

   void setup()
   {
      Serial.begin(9600);
      delay(100);
   }

   void loop()
   {
     state = 0;
      while (Serial.available()>0){
       serialByte=Serial.read();
        if ((serialByte=='S') || (serialByte == '3')){
           skinSensorBucle();

         } else if ((serialByte=='E') || (serialByte == '2')){
           ECGBucle();

         } else if ((serialByte=='B') || (serialByte == '4')){
           EMGBucle();

         } else if ((serialByte=='T') || (serialByte == '1')){
           temperatureBucle();
         }
      }
   }

   void mainMenu(void)
   {
      initialSpace();
      lineFeed();
   }

   void skinSensorBucle(void) {
     int i;
     float c=0;
     for (i=0;i<20;i++){
       float conductanceVol = eHealth.getSkinConductanceVoltage();
       Serial.print(c/2, 2);
       Serial.print(" ");
       Serial.print(conductanceVol, 2);
       Serial.println("");
```

```
      c++;
      delay(500);
    }
    mainMenu();
  }

  void ECGBucle(void) {
    int i;
    float c=0;
    for (i=0;i<1000;i++){
      float ECG = eHealth.getECG();
      Serial.print(c/67, 2);
      Serial.print(" ");
      Serial.print(ECG, 2);
      Serial.println("");
      c++;
      delay(15);}
    mainMenu();}

  void EMGBucle(void)
  {
    int i;
    float c=0;
    for (i=0;i<200;i++){
      int EMG = eHealth.getEMG();
      Serial.print(c/10, 2);
      Serial.print(" ");
      Serial.print(EMG);
      Serial.println("");
      c++;
      delay(100);
        }
      mainMenu();
  }

  void temperatureBucle(void)
  {
    int i;
    for (i=0;i<6;i++){
      float temperature = eHealth.getTemperature();
      Serial.println(temperature, 2);
      delay(1000);
    }
    mainMenu();
  }

int freeRam () {
  extern int __heap_start, *__brkval;
  int v;
  return (int) &v - (__brkval == 0 ? (int) &__heap_start : (int) __brkval); }
```