Projektiranje elemenata pametne odjeće za praćenje signala srčanog pulsa

Veldić, Marija

Master's thesis / Diplomski rad

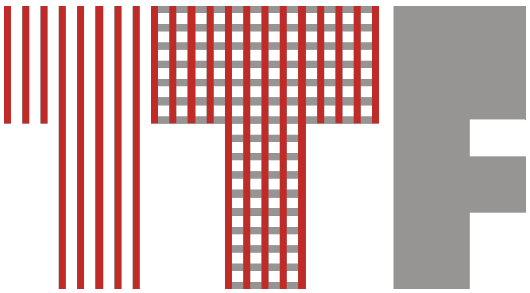
2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Textile Technology / Sveučilište u Zagrebu, Tekstilno-tehnološki fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:201:210094

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**



Repository / Repozitorij:

Faculty of Textile Technology University of Zagreb - Digital Repository







SVEUČILIŠTE U ZAGREBU
TEKSTILNO-TEHNOLOŠKI FAKULTET

DIPLOMSKI RAD

PROJEKTIRANJE ELEMENATA PAMETNE ODJEĆE ZA PRAĆENJE SIGNALA SRČANOG PULSA

MARIJA VELDIĆ

Zagreb, rujan 2017.

SVEUČILIŠTE U ZAGREBU
TEKSTILNO-TEHNOLOŠKI FAKULTET
ZAVOD ZA ODJEVNU TEHNOLOGIJU

DIPLOMSKI RAD

PROJEKTIRANJE ELEMENATA PAMETNE ODJEĆE ZA PRAĆENJE SIGNALA SRČANOG PULSA

prof. dr.sc. Dubravko Rogale

Marija Veldić, TTI/IDO 10149

Zagreb, rujan 2017.

Zahvale

Zahvaljujem se svom mentoru prof. dr. sc Dubravko Rogale koji mi je omogućio svu potrebnu opremu i pomogao sa savjetima pri izradi diplomskog rada. Također se zahvaljujem izv. prof. dr. sc. Snježana Firšt Rogale na pružanoj pomoći, strpljenju i vremenu.

Posebnu zahvalnost iskazujem kolegi Martinu Tomaševiću koji mi je pomagao pri programiranju programa za ovaj diplomski rad.

Najveću zaslugu za sve što sam postigla pripisujem svojoj mami bez koje sve ovo ne bi bilo moguće. Hvala ti na strpljenju i na neizmjerno podršci!

Na kraju bi htjela nadodati da ovaj rad posvećujem svom pk.tati bez kojeg ne bi znala koliko znači biti sretan i uspješan.

TEMELJNA DOKUMENTACIJSKA KARTICA:

Kandidat:	Marija Veldić
Naslov rada:	Projektiranje elemenata pametne odjeće za praćenje signala srčanog pulsa
Naziv studija:	Tekstilna tehnologija i inženjerstvo (TTI)
Naziv smjera:	Industrijski dizajn odjeće (IDO)
Mentor:	prof. dr. sc. Dubravko Rogale
Neposredni voditelj	izv. prof. dr. sc. Snježana Firšt Rogale
Članovi Povjerenstva:	1. doc. dr. sc. Irena Šabarić, predsjednik 2. prof. dr. sc. Dubravko Rogale, član 3. izv. prof. dr. sc. Snježana Firšt Rogale, član 4. . prof. dr. sc. Branka Vojnović, zamjenik člana
Rad sadrži:	59 stranica 22 slika 37 literaturnih referenci 1 tablica
Institucija u kojoj je rad izrađen:	Sveučilište u Zagrebu Tekstilno - tehnološki fakultet, Zavod za odjevnu tehnologiju, Laboratorij za procesne parametre
Datum prihvaćanja teme:	26. lipnja 2017. na 9. redovitoj sjednici Fakultetskog vijeća Sveučilišta u Zagrebu Tekstilno-tehnološkog fakulteta u Zagrebu
Datum predaje:	26. rujna 2017.
Datum obrane:	28. rujna 2017.

SAŽETAK

Pametna odjeća je vrsta odjeće koja ima ugrađene električke komponente i uređaje namijenjene zabavnim ili modnovizualnim izričaju, mikroračunalo te zaslon čime se omogućava komunikacija između odjevnog predmeta i nositelja takve odjeće i/ili okolišem. Integrirane elektroničke komponente omogućavaju, između ostalog, praćenje i motrenje fizioloških funkcija nositelja pametne odjeće. Takva vrsta odjeće pruža spoj multidisciplinarnih znanosti.

Za potrebe ovog diplomskog rada bit će projektirana pametna odjeća za praćenje i motrenje signala srčanog pulsa, a namijenjena je za bolesnike, rekonvalescenate, sportaše, vojnike, vatrogasce i slično. Također može pomoći njihovim liječnicima, trenerima i sl. da u realnom vremenu prate zdravstveno stanje ili ih naknadno analiziraju.

U teoretskom dijelu ovog diplomskog rada dat je osvrt na pametnu odjeću kao i na komponente koje su u nju integrirane. Prikazani su senzori koji se mogu integrirati u odjeću radi motrenja vitalnih funkcija te su opisani komunikacijski sustavi koji povezuju komponente pametne odjeće. Također su izdvojeni primjeri, na tržištu već postojeće, pametne odjeće za motrenje vitalnih funkcija kod kojih je najznačajnije motrenje signala srčanog pulsa.

U Metodici rada opisani su korišteni elementi pri realizaciji pametne odjeće za praćenje signala srčanog pulsa. Opisani su programi te uređaji koji su potrebni za dobivanje informacije o otkucaju srca.

U Eksperimentalnom dijelu i rezultatima je opisano programiranje mikrokontrolera pametne kape za praćenje signala srčanog pulsa, projektiranje pametne odjeće za praćenje signala srčanog pulsa na primjeru kape te ispitivanje senzora osciloskopom.

Na kraju diplomskog rada je Rasprava i Zaključak te popis literaturnih referenci.

Ključne riječi: pametna odjeća, signal srčanog pulsa, senzor, Lilypad Simblee BLE, bluetooth, Arduino

ABSTRACT

Smart clothing are wearables embedded with electrical components and devices intended for aesthetic use or wearables used in sport, military and health applications. Smart clothing usually consists of a sensor, a microcomputer and a screen, which enables communication between the wearers of such clothing and the environment. The integrated electronic components enable, among others, the monitoring of physiological functions of the wearer. Such wearables are a combination of multidisciplinary sciences that are bound and incorporated in such technology. The objective of this thesis is to create a heart rate pulse monitoring system. The use of such wearables is intended for patients, reconditioned athletes, soldiers, fire fighters, as well as their doctors, trainers, etc. for real-time monitoring and analysing.

The first part of the thesis is a review of smart clothing and the components used to describe it. The aim of the theoretical part is to explain and display the definition of smart clothing and what symbolizes it. There is a description of sensors that can be used as biosensors for monitoring vital signs such as heart rate, respiration rate, temperature etc. There is a given representation of the communication systems used in wearables, as well as the power supply.

The second section of the thesis describes the elements and programs used in the realization of the prototype for monitoring heart rate. Examples of the prototypes are shown, as well as a discussion given of the results and a conclusion.

Key words: smart clothing, heart rate, Lilypad Simblee BLE, bluetooth, Arduino

SADRŽAJ

1. UVOD 1

2. TEORETSKI DIO 2

2.1 POJAVA I RAZVOJ PAMETNE ODJEĆE 2

2.2 SENZORI 5

2.2.1 Primjena i tipovi senzora za biometrijska mjerenja 7

2.3 OSTALI KLJUČNI ELEMENTI U PROJEKTIRANJU PAMETNE ODJEĆE 9

2.3.1 Komunikacija unutar pametne odjeće i između udaljenih sustava 10

2.3.2 Napajanje pametne odjeće potrebnom električnom energijom 13

2.4 PAMETNA ODJEĆA ZA PRAĆENJE I MOTRENJE SIGNALA SRČANOG PULSA 14

2.4.1 Smart shirt™ 15

2.4.2 Life Shirt ™ 16

2.4.3 Wealthy odjevni sustav 17

3. METODIKA RADA 20

3.1 ARDUINO PLATFORMA 20

3.2 SIMBLEE 21

3.3 LILYPAD SIMBLEE BLE 21

3.4 SENZOR OTKUCAJA SRCA 21

3.5 OSTALE KOMPONENTE 22

3.6 MJERENJE ELEKTRIČNOG SIGNALA POMOĆU OSCILOSKOPA 23

4. EKSPERIMENTALNI DIO I REZULTATI 25

4.1 PROGRAMIRANJE MIKROKONTROLERA PAMETNE KAPE ZA PRAĆENJE SIGNALA SRČANOG PULSA 25

4.2 PROJEKTIRANJE PAMETNE KAPE ZA PRAĆENJE SIGNALA SRČANOG PULSA 30

4.3 ISPITIVANJE SENZORA OSCILOSKOPOM 32

5. RASPRAVA 35

6. ZAKLJUČAK 37

7. REFERENCE 38

Popis slika:

Sl. 1 Spoj multidisciplinarnih istraživanja u pametnoj odjeći [5] 4
Sl. 2 Fleksibilnost elektroničkih komponenata omogućuju lakšu ugradnju u pametnu odjeću [13] 10
Sl. 3 Prikaz mreže za bežičnu komunikaciju i umrežavanje svih ugrađenih komponenti u odjeći (FAN) [17] 12
Sl. 4 Osnovna mreža za trajno praćenje i bilježenje vitalnih funkcija ljudskog organizma (BAN) [36] 12
Sl. 5 Komunikacija pametne odjeće s udaljenim sustavima[21] 12
Sl. 6 Solarne ćelije ugrađene u odjevni predmet [23] 14
Sl. 7 Minijaturni električni generatori ugrađeni u obuću [24] 14
Sl. 8 Odjevna matična ploča: a) Povezivanje ORMP za različite potrebe; b) Temeljna građa ORMP [1] 16
Sl. 9 Izgled Life Shirt™ i komunikacija računalom [26] 17
Sl. 10 Pregled modula u Wealthy odjevnom sustavu [27] 18
Sl. 11 Lilypad Simblee BLE [32] 20
Sl. 12 Senzor otkucaja srca i prikaz amplitude IBI-a 21
Sl. 13 Ostale korištene komponente: a) FTDI priključak; b) Nosač baterije/ baterija 3V/ c) litij-ionska baterija 3.7V; d) LilyPad LE dioda; e) Konduktivni konac [32] 22
Sl. 14 Osciloskop oznake PicoScope [34] 23
Sl. 15 Sučelje Arduino-a na kojem je instaliran Simblee baza podataka 26
Sl. 16 Simblee aplikacija 28
Sl. 17 Simblee aplikacija povezana sa mikrokontrolerom Bluetooth tehnologijom 28
Sl. 18 Povezivanje Lilypad Simblee BLE, senzor i baterije pomoću tzv. „klema krokodilki“ 27
Sl. 19 Pametna kapa za praćenje signala srčanog s integriranim elementima 30
Sl. 20 Ispitivanje senzora osciloskopom koristeći bateriju od 3V 32
Sl. 21 21 Prikaz mjerenja srčanog pulsa starije osobe na osciloskopu 32
Sl. 22 Prikaz mjerenja srčanog pulsa mlađe osobe na osciloskopu 33

Popis tablica:

Tab. 1: Podjela skupina senzora pogodnih za pametnu i inteligentnu odjeću 6

1. UVOD

Odjeća je jedna od najstarijih uporabnih predmeta koja je nastala kao potreba čovjeka da zaštiti svoje tijelo od štetnih vanjskih utjecaja. To se najviše odnosilo na različite klimatske utjecaje kao što su hladnoća, kiša, vlaga, vrućine i slično. S vremenom je odjeća postala znak civiliziranosti i udobnosti. Koristila se u funkciji mehaničke zaštite od udaraca i ozljeda zanatlija, poljodjelaca i ratnika. Kasnije je postala znak pripadnosti nositelja ili hijerarhijskog ranga unutar organizacijskih skupina kao što su vladari, vojska i svećenstvo. Međutim, tek se kasnije odjeća počela koristiti kao estetska funkcija uljepšavanja izgleda. S početkom 20. stoljeća značajan su čimbenik odijevanja različiti materijali, seksualna orijentacija, socijalni status, tradicija, religija i tehnologija. U drugoj polovici 20. stoljeća, nakon što se počela provoditi tehnološka minijaturizacija jednostavnih elektroničkih naprava, senzora, tipkovnica, baterija i mnogih drugih komponenata, postepeno se počela razvijati i odjeća koja je sadržavala komponente i elemente koje konvencionalna i svakodnevna odjeća nije.

Pojava takve vrste odjeće s integriranim električkim i elektroničkim komponentama ubrzo je dobila naziv e-odjeća (engl. E-clothing) [1]. Nadogradnjom mikroračunala i zaslona, uz postojeće elemente e-odjeće, i omogućavanje dvosmjerne komunikacije nositelja odjevnog predmeta i okoliša, nastala je odjeća koja je uspješno obavljala pametne zadatke. Dakle, takva odjeća dobila je naziv pametna odjeća (engl. Smart clothing). Početkom 2000. godine počelo se govoriti i o takozvanoj inteligentnoj odjeći (engl. Inteligent clothing). Svojim obilježjima daleko je nadmašila konvencionalnu odjeću, e-odjeći i pametnu odjeću. Takva odjeća, danas, sadržava elemente umjetne inteligencije, a pomoću mjera i analiza parametra okoliša može samostalno donositi odluke.

Integracija multifunkcionalnih vrijednosti u odjeći, koja je neizostavni dio naše svakodnevice, postaje posebno područje velike zainteresiranosti tehnologa i odjevnih inženjera. Odjeća je postala vrlo značajna platforma za high-tech inovacije 21. stoljeća, a budućnost leži u razvoju tekstilnih i odjevnih proizvoda koji će sadržavati interakciju između pametnog tekstila i odjeće integrirane električnim komponentama. Projektiranje pametnog tekstila i pametne odjeće omogućuje proizvodnju visokotehnoloških proizvoda koji će sadržavati novi oblik ponašanja i funkcionalnosti. Konvencionalni odjevni predmet dobiva aktivan karakter i povećanu funkcionalnost. Poboljšanje kvalitete življenja, zaštita u vojnim i medicinskim svrhama te nesputana komunikacija samo su neke od mogućnosti koje nudi pametna odjeća.

2. TEORETSKI DIO

U ovom dijelu diplomskog rada bit će opisana pojava i razvoj pametne odjeće s naglaskom na odjevno računalo. Opisana je primjena senzora u odjeći te su prikazani senzori za biometrijska mjerenja, a posebno optički, biokemijski i mehanički senzori. Od ostalih komponenata koje se ugrađuju u pametnu odjeću, opisani su mikrokoprocesori, komunikacija unutar pametne odjeće i između nje te sustavi napajanja. Dati su već realizirani primjeri pametne odjeća za praćenje i motrenje signala srčanog pulsa poput Smart Shirt-a, Life Shirt-a i Weathly odjevnog sustava.

2.1 POJAVA I RAZVOJ PAMETNE ODJEĆE

Iako razvoj pametne odjeće traje već 30-ak godina, još uvijek nije u potpunosti prihvaćena kao svakodnevna pomoć u ljudskim aktivnostima te nije shvaćen njezin značaj. Zahvaljujući minijaturizaciji različitih ugradbenih elemenata (elektroničkih, pneumatskih i td.) razvoj pametne odjeće je sve značajniji.

2.1.1 Odjevno računalo

Pojava odjevnog računala (engl. Wearable Personal Computer - WPC), idejnog tvorca Stevea Manna, obilježava neprestanu interakciju između računala i korisnika [2]. Namijenjena je da obogaćuje korisnikovo okruženje novim informacijama koje sam korisnik ne bi mogao zamijetiti. Zahvaljujući primjeni odjevnog računala dobiva se veća količina podataka iz okruženja čime se, nositelju takvog odjevnog predmeta, povećana stvarnost (engl. Augmented reality).

Povećana stvarnost nositelja takve odjeće sastoji se od onog što vidi u svom okruženju i okolišu tako da se podaci i sve dodatne informacije korisniku projiciraju na određeni zaslon. Na taj način se odjevno računalo može smatrati pomagalom pri umnim i fizički aktivnostima, a da pri tome ne zahtjeva čovjekovu interakciju sa tipkovnicom ili pametnim telefonom.

Tijekom pojave i razvoja odjevnog računala te pojavom tzv. povećane stvarnosti postavljeni su i niz zahtjeva za WPC. Kao što je već spomenuto, mora omogućiti primanje naredbe na način da su ruke korisnika slobodne (engl. Hands free) ili s vrlo ograničenom uporabom ruke, odnosno prstiju. Prepoznavanje govora najpovoljniji je način povezivanja čovjeka i odjevnog računala. Također, mora biti potpuno upotrebljiv tijekom nošenja i dok je nositelj u pokretu. Mora se sastojati od senzora i komunikacijskih uređaja koji će zaprimiti podatke i izvršiti obradu istih. Jedna od najznačajnijih karakteristika odjevnih računala je njihova neprestana

interakcija sa nositeljem, pri čemu nije potrebno uključivati odnosno isključivati njegov rad čak i kada korisnik ne upotrebljava odjevni predmet.

Istraživanjem razvoja pametne odjeće, mnogi će se znanstvenici prije svega osvrnuti na Stevea Manna [3]. Njegova istraživanja i projekti doprinijeli su „(…) mogućnostima rješavanja tehničkih problema nesmetane uporabe računala i njihovih periferijskih jedinica u ljudi koji su u pokretu.“ [2]. Trenutno se mnogi istraživači na području odjevne tehnologije bave istraživanjem i izradama prototipova.

Steve Mann kao autor naziva *Wearable computer* za računala ugrađena u odjeću, uvodi još jedan termin koji se veže za njegovo ime, a to je upravo termin *Smart clothing* odnosno pametna odjeća.

2.1.2 Pametna odjeća

Osnovni elementi potrebni za izradu pametne odjeće prisutni su već više od 1000 godina. Vodljive niti poput metalnih folija zlata i srebra dugi niz godina koriste se za omatanje efektnih pređa, čime se potom „uljepšavaju“ odjevni predmeti. Do kraja 19. stoljeća, ljudi su razvili uobičajenu naviku korištenja električnih aparata. Dizajneri i inženjeri su potom počeli istraživati način kako kombinirati električnu struju i odjeću. 1988. godine vrlo je značajna prekretnica u odjevnoj industriji jer se po prvi put uvodi pojam nanotehnologija. Suradnici na kalifornijskom sveučilištu izumili su elektromotor s rotorom čiji je promjer bio manji od promjera kose. Nanotehnologija postaje inovacijski pokretač u mnogim industrijama. Razvila se kao posebno područje u izradi vrlo malih dimenzija čije područje veličine seže od nekoliko mikrometara do svega nekoliko nanometara. Izvorno namijenjen telekomunikacijskoj tehnici, biomedicinskim primjenama i automatskim mjerenjima te vođenjem procesa, ubrzo je proširila svoju primjenu na tekstilnu i odjevnu industriju.

Pri spomenu nanotehnologije u tekstilnoj i odjevnoj industriji valja ukratko spomenuti i razliku između pametnog tekstila i pametne odjeće. Pametni tekstil sastoji se od nantehnoloških obrada. Inženjeri svoja istraživanja temelje na izradi tekstilnih materijala, odnosno proizvodnji vlakana koji na svojoj površini sadrže naslojene nanočestice. Svrha nanotehnološke obrade tekstilnih materijala je mogućnost filtriranja štetnih tvari u zraku, bakterija ili otrovnih plinova. Naslojavanjem vlakana nanočesticama proizvedene su aktivne površine koji blokiraju UV zračenje, sadrže antistatička svojstva te iskazuju efekt samočišćenja. Budućnost pametnog tekstila i nantehnološke obrade leži u otkrivanju štetnih tvari u našem okruženju pri čemu će doći do eventualne promjene u boji materijala [4].

U pametnu odjeću, za razliku od pametnog tekstila, integrirane su električne i elektroničke komponente poput senzora, mikroračunala i sklopova. Nanotehnološki senzori, u pametnoj odjeći, zaslužni su za precizna mjerenja fizikalnih veličina, odnosno za mjerenja parametara unutar tekstila, fizioloških parametara tijela nositelja i/ili okoliša itd.

Prema vrsti i namjeni za ugradnju u pametnu odjeću razlikuju se mehanički, akustični, biološki i optički senzore te senzori za motrenje okoliša.

Pametna odjeća definirana je kao "(…) svojstvo odjevnog predmeta koji osigurava interaktivnu reakciju koristeći signal osjetila, obradu informacija i poticanje odziva „ [1]. Pojavila se krajem 20. stoljeća kao nova vrsta odjeće koja je omogućila dvosmjernu komunikaciju s nositeljem odjevnog predmeta i okolišem. Koncepciju pametne odjeće potaknula je ideja odjevnog računala (engl. Wearable PC, Stevea Manna), a razdoblje razvoja pametne odjeće može se razmatrati kroz četiri stupnja.

Prvi stupanj razvoja pametne odjeće pretežito se bavio razvojem i samoj integraciji već tada minijaturiziranih komponenti u odjeći. Međutim, rezultat svih ispitivanja i integriranja elektroničkih i drugih komponenti u odjevne predmete nisu postigle komercijalizaciju. Ishod istraživanja u ovom razdoblju razvoja doprinijelo je stvaranju prenosive (engl, portable) naprave umjesto nosive.



Sl. 1 Spoj multidisciplinarnih istraživanja u pametnoj odjeći [5]

Tek u drugom stupnju razvoja pametne odjeće, zapravo združivanjem tehnologije i modnog sektora odjevne i tekstile industrije, počele su se pojavljivati mnogi suradnički projekti između proizvođača elektroničkih naprava i modne industrije. U trećem stupnju razvoja pametne odjeće, marketinški razvoj počeo je privlačiti sve veću pozornost. Počeo se mijenjati pristup te su se proizvođači okrenuli potrebama korisnika. Dobili smo bolju pokretljivost i lagodnost nošenja naprava, dok su se tehničke funkcije samo unapređivale. U četvrtoj i posljednjoj fazi razvoja pametne odjeće uočavamo tržišnu zrelost i veliku zainteresiranost velikih modnih brandova. Pametna odjeća počela se upotrebljavati i koristiti u vojne, zdravstvene, sportske i rekreativne svrhe. Mikroračunala i zasloni, najčešće vizualnim ili zvukovnim načinom, postala su zaslužna za međusobnu komunikaciju između okoliša i nositelja. Prvi je put omogućena podatkovna i upravljačka komunikacija između ugrađenih komponenti, uređaja i sustava koji se nalaze u okolišu odjevnog predmeta i naravno samog nositelja [1].Pametna odjeća se može okarakterizirati i kao funkcionalna odjeća. Ovisno o sklonosti istraživača i potrebi korisnika, pametni odjevni predmeti mogu se različito interpretirati. Također, se smatra objektom multidisciplinarnog istraživanja pri čemu se nalazi na samom sjecištu područja istraživačkog dizajna, tekstilne tehnologije i fiziologije, sl. 1.

2.2 SENZORI

Senzori koji su ugrađeni u pametnu odjeću povezuju se s ostalim elektroničkim komponentama ugrađenim u odjeću, a zadatak im je prikupljanje informacija o stanju u neposrednom okolišu nositelja takve vrste odjeće, na samoj odjeći ili na tijelu nositelja. Senzori svoj rad temelje na fizikalnim i kemijskim zakonitostima. Već sada postoje preko 350 vrsti senzora koji se mogu upotrijebiti u tekstilnoj odnosno odjevnoj industriji a razlikuju se prema svojstvu transformiranja signala. Senzori se definiraju kao uređaji koji mjere fizički kvantitet i pretvaraju ga u signal kojeg može očitati promatrač ili instrument. U najširem smislu, senzori su elektroničke komponente čija svrha je otkrivanje promjena u svojoj okolini i prijenos podataka drugoj komponenti, najčešće računalu.

U tab. 1 prikazana je podjela skupina senzora pogodnih za pametnu i inteligentnu odjeću [1].

U skladu sa šest energetskih oblika postoje sljedeća područja primjene i mjerenja za [1]:

- mehaničke parametre – mjerenja duljina, površina, volumena, zakretnog momenta, pritiska, linearne i kutne brzine, akceleracije, protoka, sile, akustične valne duljine i intenziteta;

- kemijske parametre – elektrokemijska mjerenja, potenciometrijska, mjerenja koncentracije, spektroskopije, enzimatske selektivnosti, poluvodička osjetila, kalorimetrija;
- toplinske parametre – temperatura, entropija, toplinski tok, relativna vlažnost;
- električne parametre - mjerenja napona, struje, količine naboja, otpora, induktiviteta, kapaciteta, dielektrične konstante, polarizacije, električnog polja, frekvencije, dipolnog momenta;
- magnetske parametre – jakost polja, gustoća magnetskog toka, magnetski moment, permeabilnost;
- parametara elektromagnetskog vala - intenzitet, faza, valna duljina, polarizacija, refleksija, transmisija, indeks loma.

Tab. 1: Podjela skupina senzora pogodnih za pametnu i inteligentnu odjeću

Tip senzora	Mjerna veličina	Korištenje u inteligentnoj odjeći
mehanički	pozicija, sila, brzina, ubrzanje, masa, pomak, oblik	detekcija ljudi i predmeta u prostoru, određivanje pozicije, težine kretanja
akustički	intenzitet zvuka, frekvencija, faza	detekcija zvuka, interpretacija govora
biološki	srčani puls, tjelesna temperatura, krvni tlak, moždana aktivnost (EEG), srčana aktivnost (EKG), respiracija	mjerenje zdravstvenog stanja tijela, mentalnog stanja, fizičkog stanja i aktivnosti
optički	intenzitet svjetla, frekvencija svjetla, ogib	računalna vizualizacija prostora, infracrvena detekcija nazočnosti i gibanja
motrenje okoliša	temperatura zraka, vlažnost zraka, brzina strujanja zraka, prozirnost zraka	mjerenje stanja u okolišu u kojem se nalaze ljudi

Količina senzora unutar pametne odjeće može varirati ovisno o kompleksnosti sustava mjerenja unutar odjevnog predmeta, odnosno o namjeni pametnog odjevnog premeta. Odjevni predmet može sadržavati nekoliko različitih senzora i njegovih pretvornika. Na primjer, elektroakustični senzor za ulazni i izlazni zvuk; elektrokemijski za napajanje; elektromagnetski za bežičnu komunikaciju; elektromehanički za osjetljivost na pritisak i termoelektrični za upravljanje i praćenje temperature [6].

2.2.1 Primjena i tipovi senzora za biometrijska mjerenja

Unutar strujnog kruga, senzori su poput ljudskih osjetila. Kao što čovjek prima informacije iz svoje okoline pomoću svojih osjetila, na isti se način ponašaju i senzori. Ugradnja senzora u pametnu odjeću omogućuje prilagodbu njihovih funkcija kako bi bolje odgovarali aktivnostima nositelja. Senzori koji se koriste za biometrijska mjerenja služe se određenim biološkim elementom. Takvi senzori omogućavaju interakciju između pametnog odjevnog predmeta i nositelja, a njihov rad se temelji na algoritmu koji predviđa i otkriva opasnosti izvan zadanog omjera [7]. Biološki element može biti biofizički rezultat ili spoj poput znoja. Senzori svoj rad zasnivaju na analizi područja interesa kako bi proizveli učinak koji je mjerljiv. U pametnoj odjeći senzori su komponente koje se mogu nositi na tijelu u obliku nosivih uređaja ili mogu biti integrirani u sam odjevni predmet. Senzori se mogu koristiti za npr. neprekidno i kontinuirano očitavanje krvnog tlaka, srčanog pulsa, tjelesne temperature i sličnih vitalnih funkcija nositelja pametne odjeće. Biosenzori se danas razvijaju kako bi poboljšali kvalitetu života i prevenirali moguće zdravstvene komplikacije.

2.2.1.1 Optički senzori

Optički senzori koriste mogućnost pretvorbe električnog signala u svjetlost (LED) i mogućnost promjene vodljivosti poluvodičkog materijala obasjanog svjetlošću (fotodiode i fototranzistori). Za emitiranje svjetlosti pomoću LED služi odašiljač (engl. emitter), a za primanje koristeći fotodiodu ili fototranzistor prijemnik svjetlosti (engl. receiver). Tri su osnovne vrste optičkih senzora [8]:

- difuzijski (engl. diffuse sensors)
- retro-reflektirajući (engl. retro-reflective sensors) i
- senzori s prolaznom zrakom (engl. through-beam sensors)

Difuzijski senzori detektiraju svjetlost reflektiranu od objekta kojeg se treba detektirati, a retro-reflektirajući i senzori s prolaznom zrakom detektiraju objekt ako on spriječi put zrake svjetlosti od odašiljača do prijemnika.

Optički senzori pružaju samo informacije koje se odnose na svjetlosnu razinu, odnosno na električni otpor koji se javlja na komponenti foto-otpornika (engl, light dependant resistor LDR). Foto-otpornik, u ovom slučaju, je otpornik čiji se električni otpor smanjuje s povećanjem intenziteta ulazne svjetlosti. Ako bi se foto-otpornik koristio kako bi aktivirao svjetiljku, kada se smanji intenzitet prirodnog svjetla, analogni ili digitalni sklop će izvesti aktivaciju svjetiljke. Razlika između analognog i digitalnog sklopa je da digitalni sadrži vrijednosti veličine između 0 do 255, a analogni se služe podesivim brojčanikom. U digitalnom sustavu, pad intenziteta svjetlosti za 50% označio bi vrijednost 128. U tom slučaju 0 predstavlja potpuni mrak dok 255 označava potpuno osunčanu atmosferu [6]. Najjednostavniji primjer pametne odjeće s ovakvim senzorom može se koristiti za uočavanje pješaka na cesti u noćnim uvjetima. U padu intenziteta svjetlosti, svjetiljke na odjevnom predmetu se uključuju i time nosač takvog odjevnog predmeta postaje vidljiv. S obzirom da se ovo istraživanje temelji na pametnoj odjeći za mjerenja vitalnih funkcija u medicinskim svrhama, opisani senzor je od bitne važnosti jer se pomoću optičkog senzora može izvoditi računalna vizualizacija prostora, infracrvena detekcija nazočnosti i gibanja i sl.

Provedena su značajna istraživanja u vezi s optičkim senzorima u pametnoj odjeći koji omogućava prijenos podataka na pametni telefon. Dokazano je da se pomoću senzora može pouzdano pratiti broj otkucaja srca te zasićenost krvi kisikom [9]. Optički senzori sastoje se od pretvornika i LE dioda koji kada osvijetle površinu kože mjere razliku u količini reflektirane svjetlosti koja nastaje zbog pulsiranja krvnih žila.

2.2.1.2 Biosenzori

U literaturi se vrlo često isprepliću pojmovi kemijskog i biokemijskog senzora, tj.biosenzora. Biosenzori su podgrupa kemijskih senzora gdje je biološki aktivan materijal: enzim, antitijelo, biološka kemikalija, tkivo, stanica i sl. [10]. Mjerne veličine koje se mogu mjeriti pomoću ove vrste senzora su srčani puls, tjelesna temperatura, krvni tlak, moždana aktivnost (EEG), srčana aktivnost (EKG), respiracija, odnsono može se izvoditi mjerenje zdravstvenog stanja tijela, mentalnog stanja, fizičkog stanja i aktivnosti nositelja odjeće u koju je integriran optički senzor. Posljednjih desetak godina biosenzori su postali vrlo značajni u svijetu nosive tehnologije. Nastali su kao privremene tetovaže na površini kože čiji rad se temelji na potenciometrijksim

mjerenjima. Senzor vezan uz minijaturni bežični primopredajnik prati količinu natrija u ljudskom znoju. Dokazano je da mjerenjem količine natrija u znoju možemo predvidjeti neravnotežu elektrolita, što je dobar pokazatelj općeg zdravlja kod svakog čovjeka. Također su vrlo korisni s obzirom da imaju mogućnost motrenja radne okoline štetne po ljudskom zdravlju. Brzim napretkom u razvoju nanotehnologije, biosenzori velikim tempom postigli su značajne rezultate integracije u pametnu odjeću [9].

2.2.1.3 Mehanički senzori

Mehanički senzori unutar pametne odjeće imaju vrlo široku primjenu u pametnoj odjeći. Namijenjeni su za detekciju ljudi i predmeta u prostoru, određivanje pozicije, težine kretanja i sl. Kontinuirano motrenje fizičkih aktivnosti može pomoći u otkrivanju kardiovaskularnih, neuroloških i plućnih bolesti od samog početka njihovog pojavljivanja. Senzori mjere poziciju, sile, brzine, ubrzanja, masu, linearne i kutne pomake tijela. Time je moguće dobiti informacije o načinu i tempu kretanja nositelja odjevnog predmeta. Također, motrenje aktivnosti pojedinaca pomoću mehaničkih senzora, mogu biti korisni u otkrivanju kvalitete spavanja, držanju tijela kao i za detekciju pada, što je vrlo korisno starijim i nemoćnim osobama. Osim njihove integracije u pametnu odjeću, sustav namijenjen motrenju fizičkih aktivnosti pojedinaca može se nalaziti u kućnom okruženju kako bi utvrdili vrstu i intenzitet kretanja osobe.

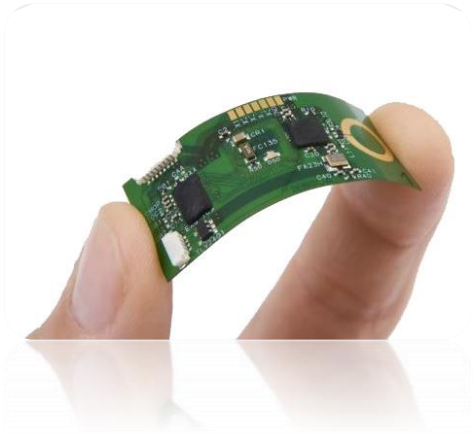
2.3 OSTALI KLJUČNI ELEMENTI U PROJEKTIRANJU PAMETNE ODJEĆE

U ovom poglavlju bit će opisani mikrokotroleri, kao ključni elementi integrirani u pametnu odjeću. Također je opisana komunikacija unutar pametne odjeće i između udaljenih sustava te sustavi napajanja komponenata ugrađenih u pametnu odjeću.

2.3.1 Mikrokontroler

Pomoću sveprisutnih mikrokontrolera i mikroprocesora povećana je pouzdanost i preciznost upravljanja pojedinim sustavom. Sustav oplemenjen mikrokontrolerom može u potpunosti zamijeniti čovjeka, smanjuje dimenzije uređaja i potrošnju energije. Zbog jednostavnosti i širokog spektra uporabe proširili su krug ljudi koji se bave pametnom i inteligentnom odjećom [11]. Mikrokontroler je malo računalo smješteno na jedan integrirani sklop unutar kojeg se nalaze procesor, radna memorija, programska memorija, ulazne i izlazne jedinice i sl. [12]. Mikrokontroler, s pripadajućim programom je obradbena jedinica za interpretaciju ulaznih informacija i donošenje odluka [1]. Fleksibilnost elektroničkih

komponenata omogućuju lakšu ugradnju u pametnu odjeću te mogućnost da se izradi ugodniji sustav za nošenje, sl. 2 [13].



Sl. 2 Fleksibilnost elektroničkih komponenata omogućuju lakšu ugradnju u pametnu odjeću [13]

Mikroprocesori su vrlo složeni sustavi koji sadrže kompleksne električne sklopove namijenjene za pohranjivanje uputa programa i obradu podataka, a bez kojih je gotovo nemoguće stvoriti dinamičan i pametan odjevni predmet. Zahvaljujući minijaturizaciji omogućena je proizvodnja mikrokontrolera sastavljena od različitih komponenti poput mikroprocesora, komunikacijskih sustava, pojačala i sl. [14] [15] [16] .

2.3.1 Komunikacija unutar pametne odjeće i između udaljenih sustava

Svjedoci smo izuzetnog napretka u području komunikacijske tehnologije, koja je vrlo važan element pametne odjeće. Kod ove vrste odjeće komunikacija se odvija:

- unutar pametnog odjevnog predmeta (komunikacija između ugrađenih komponenti),
- između pametnog odjevnog predmeta i nositelja tog odjevnog predmeta (preko mikroračunala i njegovog zaslona najčešće vizualnim ili zvukovnim načinom
- između pametnog odjevnog predmeta i sustavima koji se nalaze u okolišu odjevnog predmeta te udaljenih sustava.

Svrha komunikacije unutar pametne odjeće je izvještavanje krajnjeg korisnika (bolesnika, vojnika, sportaša, ali i liječnika, trenera i sl.)

Komunikacije mreže koje se koriste kod pametne odjeće su:

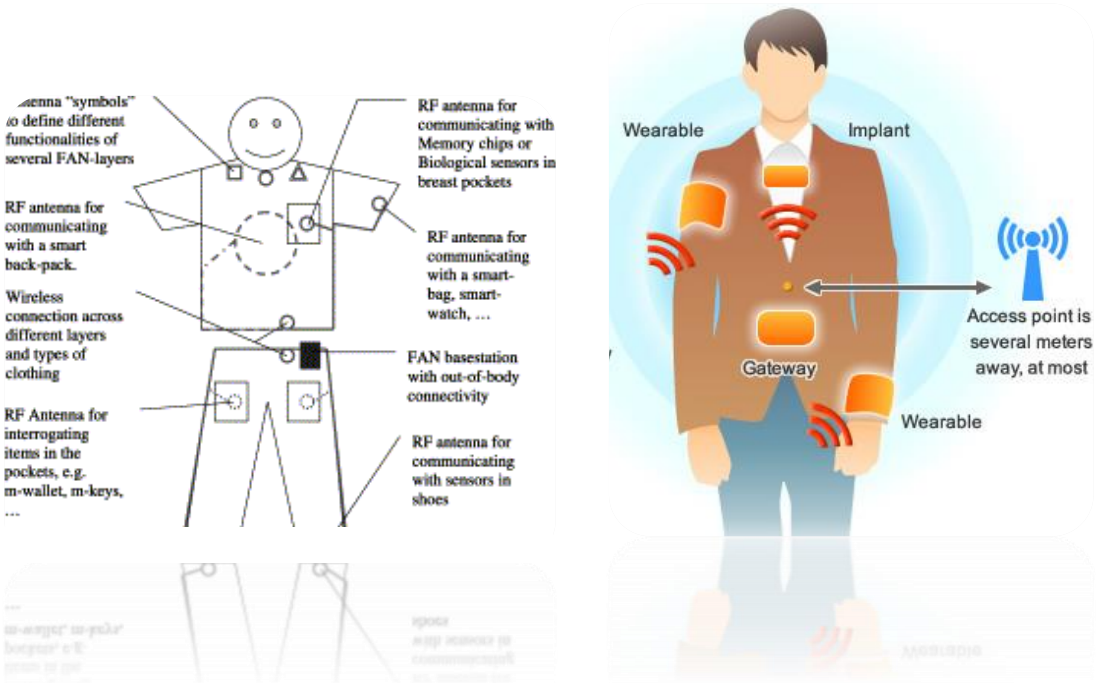
1) FAN (engl. Fabric Area Network) – mreža za bežičnu komunikaciju i umrežavanje svih ugrađenih komponenti u odjeći, sl. 3 [17].

2) BAN (engl. Body Area Network) - osnovna mreža za trajno praćenje i bilježenje vitalnih funkcija ljudskog organizma i / ili kretanja čovjeka u realnom vremenu, a omogućuje rano otkrivanje simptoma bolesti, praćenje kronično oboljelih pacijenata, nadzor pacijenata koji se oporavljaju od operativnih zahvata ili samo sigurnu rehabilitaciju unutar vlastitog doma, sl. 4 [18].

3) PAN (engl. Personal Area Network) - osobna mreža koja predstavlja računalnu mrežu najbližu ljudskom tijelu i koja obrađuje informacije iz najbližeg okoliša nositelja pametnog ili inteligentnog odjevnog predmeta. Većina autora BAN i PAN smatra ekvivalentnima [18].

4) WLAN (engl. Wireless Local Area Network) - označava lokalnu mrežu (LAN) koja se zasniva na bežičnim tehnologijama u okviru jednog ureda, zgrade, postrojenja ili kuće [19].

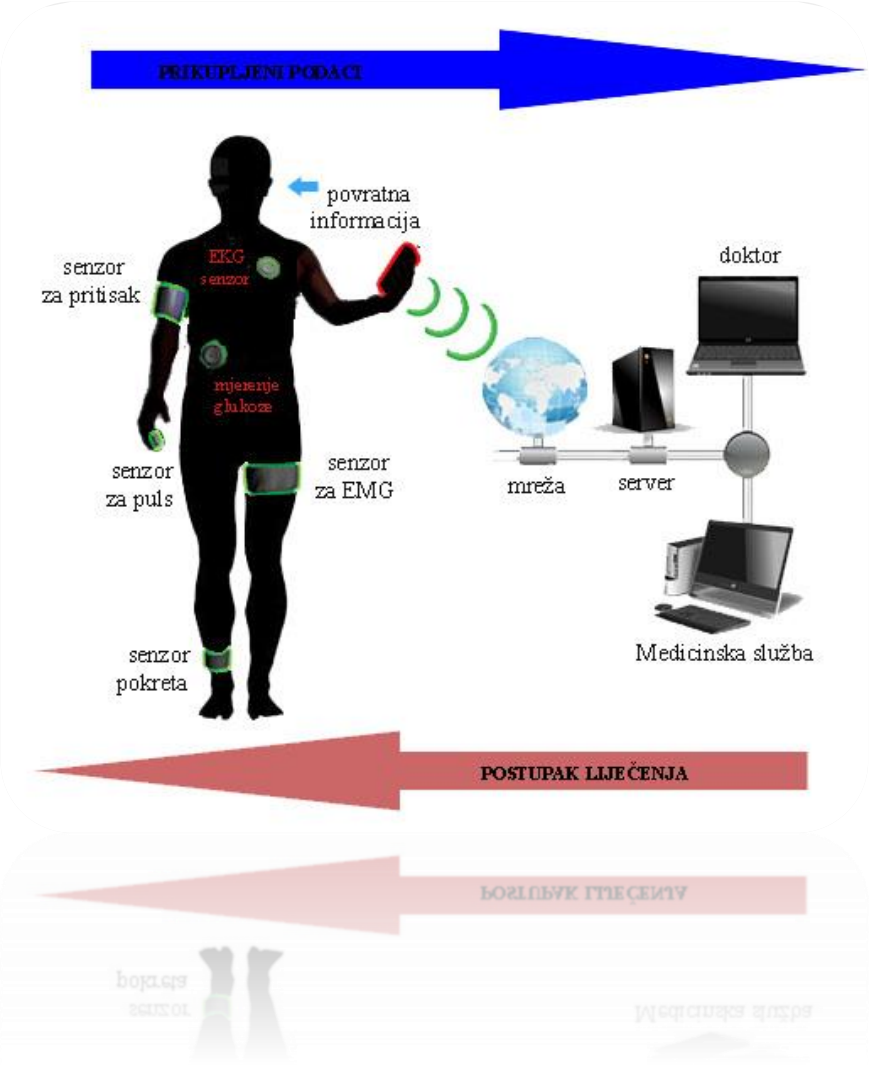
5) LAN (engl. Local Area Network) - lokalna (područna) računalna mreža za povezivanje u okviru jednog ureda, zgrade, postrojenja ili kuće [19].



Sl. 3 Prikaz mreže za bežičnu komunikaciju i umrežavanje svih ugrađenih komponenti u odjeći (FAN) [17]

Sl. 4 Osnovna mreža za trajno praćenje i bilježenje vitalnih funkcija ljudskog organizma (BAN) [36]

Najprikladniji način bežičnog povezivanja je povezivanje putem Bluetooth-a. Bluetoooth uređaji su uređaji kratkog dometa, ali vrlo sigurni i povjerljivi, koji se služe malom količinom napajanja [20]. S obzirom na ograničeni domet koristi se za povezivanje komponenata unutar pametne odjeće i za povezivanje osobnog mobitela, laptopa i sl.



Sl. 5 Komunikacija pametne odjeće s udaljenim sustavima[21]

Bežična komunikacija udaljenih sustava, sl. 5, omogućena je i uz pomoć repetitora mobilnih uređaja. Osnova sustava je bežični usmjerenik koji distribuira komunikacije uz pomoć radio valova različitih frekvencija.

2.3.2 Napajanje pametne odjeće potrebnom električnom energijom

Današnji način života zahtjeva od čovjeka da pri obavljanju svakodnevnih aktivnosti koristi sve više uređaja koji su prenosivi i imaju baterijsko napajanje. Vrlo često ti uređaji zahtijevaju veće količine električne energije nego što ih može pribaviti ugrađena baterija (nedovoljnog kapaciteta).

Uređaji integrirani u pametnu odjeću također zahtijevaju napajanje koje se može izvoditi iz više izvora. Osim baterijskog napajanja, može se koristiti ambijentalno svjetlo (izravno sunčevog ali i svjetla od umjetnih izvora svjetlosti u zatvorenim prostorima), energija radiovalova iz okolišnog prostora, vibracijski mikrogeneratori, ambijentalni protok zraka, aktiviranje tipaka na tipkovnicama i sl. Također se za napajanje elektroničkih komponenti ugrađenih u odjeću može koristiti energije hodanja ili toplinska tjelesna energija koja se pretvara u električnu energiju. Glavni problem kod odjeće s ugrađenim komponentama je upravo način napajanja svih elektroničkih elemenata. Kako stvoriti potrebnu količinu energije koja neće podrazumijevati prevelike ili ogromne baterije. U idealnoj situaciji napajanje električnih komponenti pametnih odjevnih predmeta ne bi bilo potrebno, no još uvijek se nisu dosegla takva idealna tehnološka rješenja. Prikupljanje energije sažima nekoliko različitih postupaka koji bi mogli dovesti korak bliže idealnom napajanju. Baterijsko napajanje današnje pametne odjeće, s aspekta mase predstavlja 50% težine u kojoj su ugrađene predviđene komponente takve odjeće.





Sl. 6 Solarne ćelije ugrađene u odjevni predmet[23]

Sl. 7 Minijaturni električni generatori ugrađeni u obuću [24]

Utvrđeno je da se čak i količina hodanja može pretvoriti u energiju kao i toplina tijela. Međutim, takav način dobivanja energije i dalje nije dovoljan za napajanje složenijih elemenata ugrađenih komponenti u odjeći. U mnogim slučajevima napajanja pametne odjeće koriste se komercijalne baterije. Baterije nisu dovoljno minijaturizirane te njihovo napredovanje zaostaje za novim tehnološkim otkrićima a problem nastaje jer se energija koju koriste za napajanje troši i sa svakom potrošnjom se trebaju mijenjati.

Sustav integriranih minijaturnih elektroničkih uređaja ugrađenih u pametnu odjeću mogu se, uz baterijsko napajanje, nadopunjavati i iz solarnih ćelija ugrađenih na ramene dijelove odjevnog predmeta, sl. 6, torbe ili uporabom cipela s izvorom električne energije, sl. 7. U cipele mogu biti ugrađeni minijaturni električni generatori koji proizvode električnu energiju tijekom hodanja, a proizvedena električna energija služi za nadopunjavanje sustava baterija [22].

Postoji širok spektar novih tehnologija napajanja pametne odjeće koji se danas razmatraju i proučavaju u području tehnologije i odjevne industrije.

2.4 PAMETNA ODJEĆA ZA PRAĆENJE I MOTRENJE SIGNALA SRČANOG PULSA

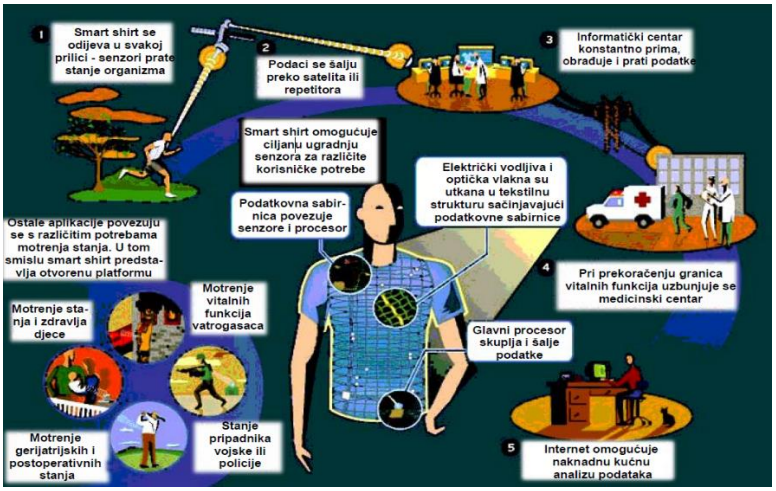
Kvalitetnija i učinkovitija zdravstvena njega, u bolnici ali i kod kuće, sve je potrebnija bolesnicima i rekonvalescentima. Osim što je pacijentima potrebna, ali i dragocjena, veća udobnost također je i vrijedna društvu kako bi se povećala zdravstvena učinkovitost. Težimo ka boljem i kvalitetnijem životu čak i onda kada patimo od kroničkih bolesti. Nosivi senzori koji se danas mogu integrirati u odjeći imaju mogućnost sprječavanja bolesti, dijagnosticiranje istih i pružaju interaktivno motrenje ključnih vitalnih funkcija. Pametna odjeća u medicinskim svrhama dat će pacijentima odgovarajuće informacije o tome kako postupiti s određenom bolesti, ukazati na fizičku neaktivnost ili nivo stresa, te pružiti eventualnu motivaciju za promjenu. Dokazano je da kroz dugoročne analize moguće je uočiti i spriječiti potencijalne zdravstvene komplikacije. Dakle, pametna odjeća namijenjena bolesnicima i rekonvalescentima vrlo je značajna jer se koristi u cilju praćenja zdravstvenog stanja ili radi provođenja određene terapije.

Postoje dva oblika pametne odjeće za motrenje i praćenje vitalnih funkcija. Prvi oblik podrazumijeva integraciju senzora koji su u doticaju s kožom. Senzori su uglavnom okruženi slojevima materijala dok u nekim slučajevima se i materijal ponaša kao senzor (piezo-resistivna pređa, optička vlakna). Dakle, pametna odjeća za biometrijsko motrenje pruža prednost kao što su: diskretnost, zaštićenost, korisniku omogućava jednostavno korištenje te razumijevanje. Drugi oblik pametne odjeće podrazumijeva korištenje senzora koji su smješteni u džepovima tj. nisu u izravnom kontaktu s kožom. Ovakvi odjevni predmeti pružaju informacije kao što detekcija pada, motre fizičku aktivnost i način kretanja te pružaju informacije o lokaciji nositelja.

2.4.1 Smart shirt™

Prvi primjer pametne odjeće namijenjene za praćenje vitalnih funkcija nositelja nastao je prije dva desetljeća u SAD-u na Georgia Institute of Technology gdje su znanstvenici razvili takozvanu Odjevnu računalnu matičnu ploču – ORMP (engl. Wearable Motherboard), sl. 8.

Razvoj odjevne računalne matične ploče prvenstveno je bio namijenjen za vojne svrhe. S obzirom da je načinjena od optičkih vlakana, pružila je mogućnost u otkrivanju mjesta postrijeljane rane u borbenim uvjetima. Istkana je tako da su polimeri optičkih vlakana i ostali specijalni vodiči integrirani u strukturu tkanine, različiti senzori se mogu fizički odvojiti i pozicionirati ovisno o potrebama nositelja odjevnog predmeta.



a)



b)

Sl. 8 Odjevna matična ploča: a) Povezivanje ORMP za različite potrebe; b) Temeljna građa ORMP [1]

Budući da je ORMP zamišljena kao podloga za ugradnju senzora, upravljačkih funkcija i računalno informatičkih naprava, ona može biti korištena u bilo koje svrhe koje zahtijevaju motrenje vitalnih životnih funkcija. [1] Smart shirt™ pruža komunikaciju između pacijenta i liječnika.Omogućuje neprestanu interakciju između pacijenta i liječnika čak i kada se pacijent upućuje kućnoj njezi ili kada se radi o pacijentima koji žive u ruralnim područjima. S obzirom da se može prilagoditi određenim tipovima tijela, koristi se i za praćenje vitalnih funkcija kod novorođenčadi kako bi se spriječio iznenadni prestanak disanja. Odjevna računalna matična ploča pokazala se vrlo učinkovita u motrenju i mentalnih oboljenja kako bi razumjeli i razjasnili uzorke ponašanja. Informacije dobivene ovim odjevnim predmetom značajno poboljšavaju kvalitetu dijagnosticiranja i pružaju informacije pomoću kojih liječnici lakše donose odluke ili poboljšavaju postupke terapije.

2.4.2 Life Shirt ™

Life Shirt, sl. 9 [25] je odjevni predmet koji, ovisno o potrebama korisnika, može izvoditi različita mjerenja vitalnih parametara (EKG, temperature, respiracije, pokreta, držanja i sl.) ali i pružiti psihosocijalnu potporu (gumb za uzbunu, dostava lijeka, prikaz lokacije pacijenta i sl.) Odjevni predmet koristi bežičnu komunikaciju te se podaci pohranjuju i analiziraju kako bi stvorili datoteke koje se prenose liječniku ili medicinskom radniku pomoću računalne mreže i

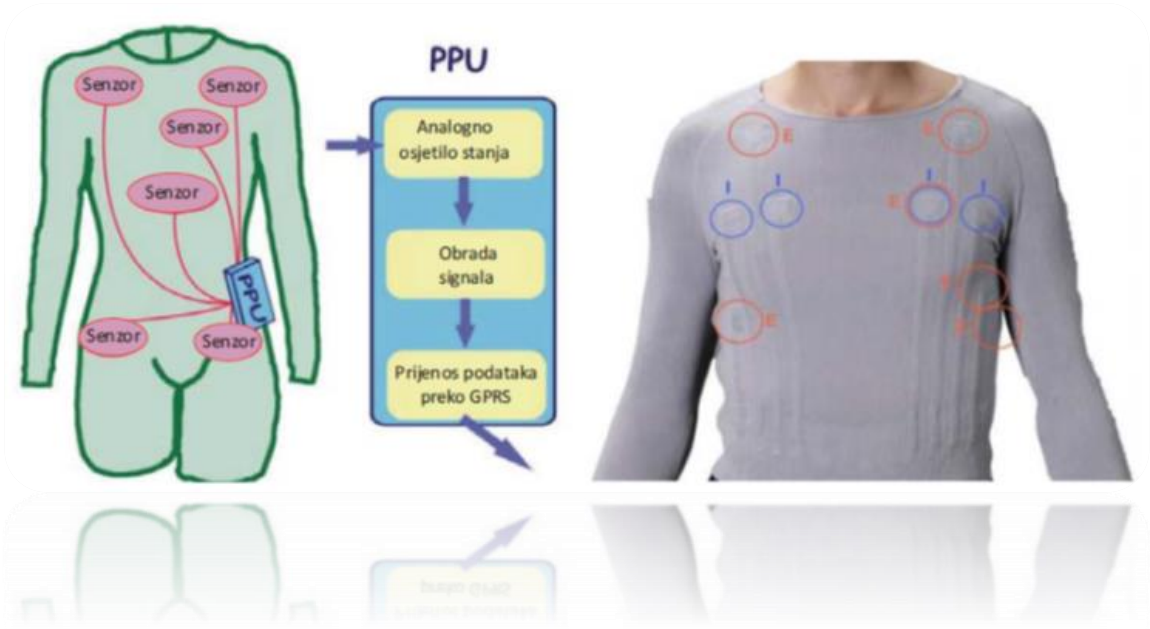
infrastrukture. Life Shirt je razvio i proizvod kojeg danas koriste i sportaši. Prati i analizira kardiovaskularni sustav tijekom treninga a putem Bluetooth tehnologije šalje podatke osobnom treneru. Istovremeno ima mogućnost prikupljanja podataka na memorijsku karticu koja se može naknadno analizirati.



Sl. 9 Izgled Life Shirt™ i komunikacija računalom [26]

2.4.3 Wealthy odjevni sustav

Wealthy odjevni sustav, sl. 10, (engl. Wealthy healthcare system) [27] sastoji se od ugrađenih senzora koji se postupkom pletenja integriraju u odjevni predmet. Zahvaljujući različitim pređama koji imaju svojstva provodljivosti te piezoresistivnim osjetilima, za detekciju pokreta, omogućena je izvedba pametne odjeće koja ima sposobnost snimanja fizioloških aktivnosti. Podaci se pomoću minijaturnih bežičnih sustava, u realnom vremenu, prenose na računalo ili liječniku. Wealthy odjevni sustav namijenjen je prvenstveno bolesnicima koji pate od srčane bolesti tijekom i nakon rehabilitacije.



Sl. 10 Pregled modula u Wealthy odjevnom sustavu [27]

Specifičnost ovakvog sustava za praćenje i motrenje vitalnih funkcija je da su senzori u obliku pređe izvedene pomoću lycre obložene gumom. Ponašanje senzora se razlikuje ovisno o svojstvu rastezljivosti prema osnovi i potki a komunikacija između komponenata je postignuta upotrebom konduktivnog konca ili poliesterskih pređa obavijeni nitima srebra. Prednost Wealthy odjevnog sustava je da su pređe otporne na pranje. Dokazano je da senzori mjere na jednak način prije i nakon njege.

3. METODIKA RADA

U ovom dijelu diplomskog rada bit će prikazani programi i komponente korištene za projektiranje elemenata pametne odjeće (kape) za praćenje signala srčanog pulsa: Arduino platforma, Bluetooth, senzora za mjerenje otkucaja srca te mikroprocesora Lilypad Simblee BLE s ugrađenim uređajem te ostale komponente. Također će biti opisan korišteni osciloskop PicoScope.

3.1 ARDUINO PLATFORMA

Arduino, računalno - softversko poduzeće osnovano 2005. godine u Italiji. Osnivači, Massimo Banzi i David Cuartielle, su razvili Arduino kao platforma bazirana na otvorenom kodu (eng. open-sorce), jer kako je M. Banzi izjavio "Ne treba vam ničija dozvola da kreirate nešto sjajno." Osnivačima je bio cilj stvoriti jednostavne, ali i cjenovno niske, elektroničke sklopove i komponente koji će biti dostupne široj masi za kreiranje interaktivnih projekata [28]. Arduino je platforma za kreiranje elektroničkih prototipova, a namijenjen je elektroničarima, dizajnerima i svima ostalima koji žele stvarati interaktivne proizvode.

Arduino mikrokontroleri su komercijalno dostupni kao gotovi pribori ili pribor za tzv. *do-it-yourself* (uradi sam) uporabu. Zajednički je naziv različitih skupinama mikrokontrolera. Ploče sadrže kombinaciju RAM memorije, flash memorije te ulazne i izlazne jedinice. Prema tome, njihova izvedba vrlo je slična računalima ali sadrže izrazito manja radna svojstva. Dio Arduino-a sastoji se i od programskog paketa koji je osmišljen tako da koristi jednostavne naredbe, a prebacivanje koda ostvaruje se pomoću USB kabela. Nakon programiranja Arduino može raditi dok je povezan na računalu ili samostalno pomoću baterije. Softverski program pod nazivom Arduino IDE (engl. Integrated development environment) sastoji se od programskog jezika poznat pod nazivom Java. Pruža jednostavne mehanizme za sakupljanje i prenošenje programa na Arudino ploče [29].

S obzirom da je Ardiuno utemeljen na tipu platforme otvorenog koda, programiranje je svedeno na najosnovnije programiranje, a upravo je to dovelo do stvaranja široke zajednice korisnika koji svakodnevno sudjeluju u pružanju i objavljivanju kodova i uputa za veliki broj Arduino projekata.

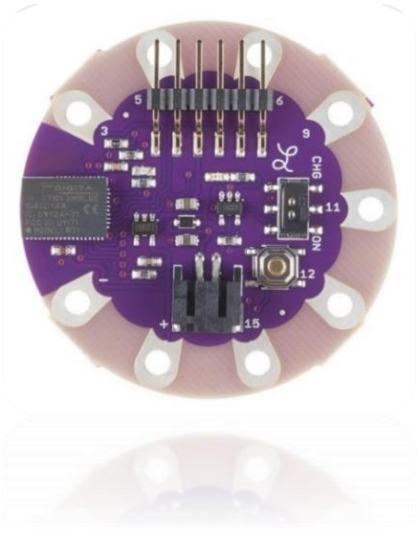
3.2 SIMBLEE

Simblee je uređaj koji se može programirati u programu Arduino IDE te se pomoću niskoenergetskog Bluetooth-a prenosi kod na Simblee aplikaciju [30]. U osnovi je namijenjen za mobilnu aplikaciju koju neće biti potrebno programirati pomoću programskog jezika namijenjen za razvoju aplikacije bilo na IOS ili Android platformi.

Simblee nudi i nekoliko sučelja za programiranje aplikacija (engl. Application programming interface, API). Najbolji i dobro dokumentirani su SimbleeForMobile, SimbleeCloud i SimbleeCOM. Svako od ovih sučelja sadrži određeni skup pravila i specifikacija koje programeri slijede kako bi se mogli služiti uslugama ili resursima uređaja. SimbleeCloud usmjeren je na slanje podataka opsluženom serveru, SimbleeCom služi za međusobnu komunikaciju između dva Simblee uređaja a SimbleeForMobile usredotočen je na stvaranju aplikacija za IOS ili Android.

3.3 LILYPAD SIMBLEE BLE

Lilypad Simblee BLE, sl. 11, je nosiva razvojna ploča koja omogućava povezivanje s pametnim telefonom putem Bluetooth-a [31]. Za razliku od ostalih BLE uređaja, Lilypad Simblee BLE ne zahtijeva stručno znanje za razvoj aplikacije. Programiranje Simblee-ja se obavlja isključivo u Arduino IDE programu te se povezivanjem računala i mikrokontrolera prenosi programirani kod.



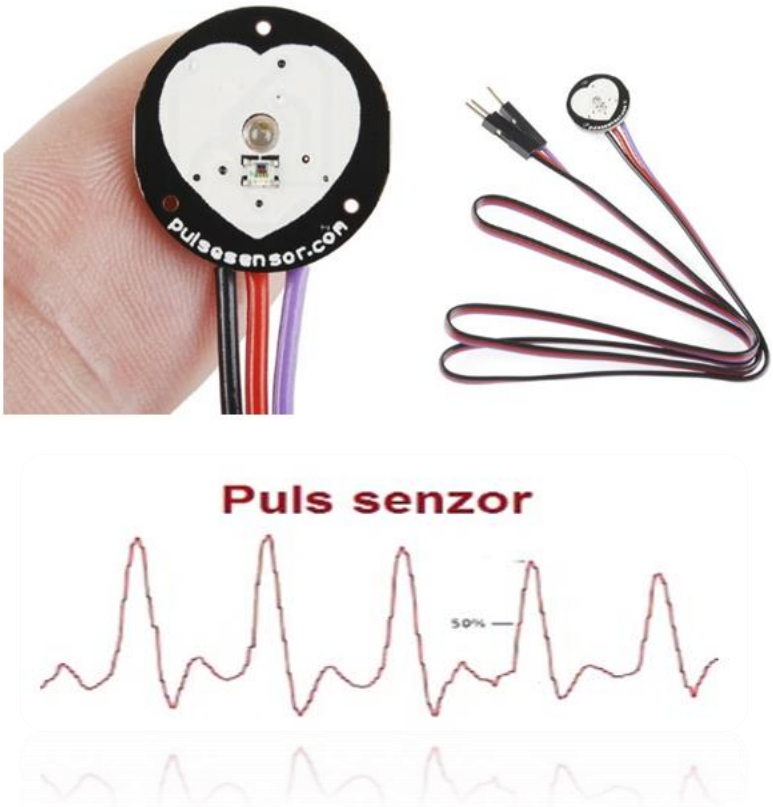
Sl. 11 Lilypad Simblee BLE [32]

Lilypad Simblee BLE je mikrokontrolerski sklop koji je namijenjen izradi interaktivne odjeće. Vrlo je svestrana tehnologija koja se danas koristi za projektiranje pametne odjeće. Osnovna ideja je stvoriti i omogućiti komunikaciju između nositelja i pametnog odjevnog predmeta.

Pričvršćivanje dodatnih komponenata koji međusobno uspostavljaju vezu izvodi se spajanjem (šivanjem) konduktivnim koncem. Modularna struktura Lilypada također omogućava povezivanje različitih senzora, baterija, LED dioda i slično.

3.4 SENZOR OTKUCAJA SRCA

Senzor otkucaja srca je tkz. *plug-and-play* senzor koji radi na Arduino platformi [33], sl. 12. Namijenjen je studentima, umjetnicima, sportašima, kreativcima te programerima koji, u svoje projekte, žele ugraditi podatke dobivene otkucajem srca. U osnovi senzora je optički senzor koji sadrži pojačalo i strujni krug zaslužan za otklanjanje šuma.



Sl. 12 Senzor otkucaja srca i prikaz amplitude IBI-a

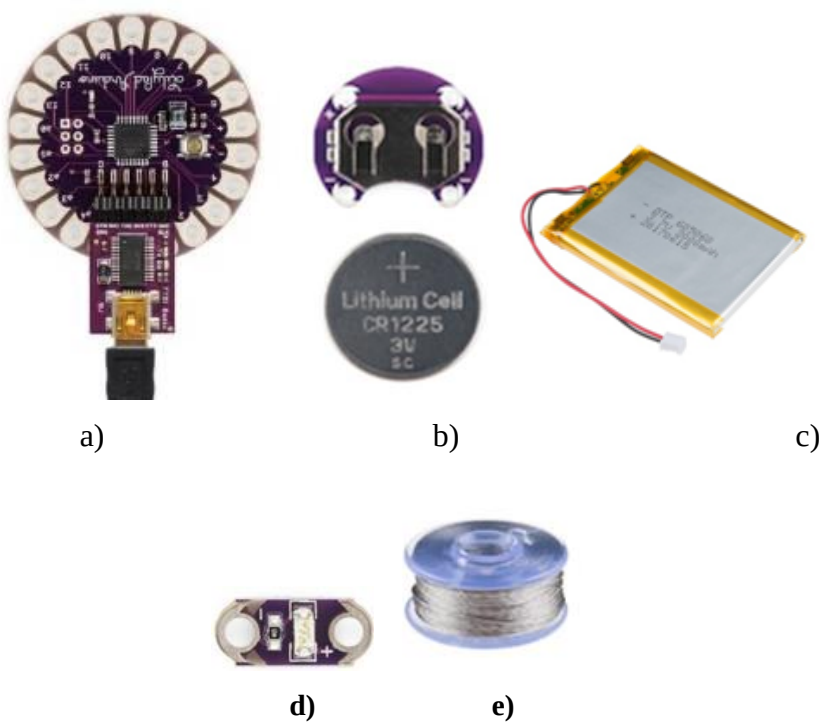
Senzor bilježi brzinu otkucaja srca u milivoltima (mV) koji se mogu prikazati pomoću fotopletmograma ili PPG-a. Pulse Senzor svoj rad temelji na električni otpor koji mjeri razliku u količini reflektirane svjetlosti na površini kože. S povećanjem intenziteta svjetlosti, signal će se povećati dok će se s manjim intenzitetom svjetla smanjiti.

Senzor mjeri brzinu otkucaja srca tako da pronalazi uzastopna razdoblja srčanog pulsa i mjeri razliku između svakog otkucaja. Razdoblje između svakog otkucaja naziva se engl. *Inter-Beat*

Interval (IBI). IBI se mjeri tako da se bilježi vrijeme u trenutku kada signal prijeđe 50% amplitude a zatim se brzina otkucaja srca pretvara u otkucaje po minuti kao prosjek prethodnih 10 IBI-a.

3.5 OSTALE KOMPONENTE

Kako bi se za potrebe ovog diplomskog rada prebacio programirani kod na mikrokontroler Lilypad Simblee BLE korišten je i tkz. Lilypad FTDI Basic Breakout priključak, sl. 13.a), koji povezuje računalo i mikrokontroler. Koristio se i nosač baterije koji sadrži dvije pozitivne i dvije negativne spojne rupice kao i baterija kružnog oblika čije napajanje seže do 3V, sl.13.b). Litij-ionska baterija od 3.7 V je korištena za napajanje Lilypad Simblee BLE mikrokontroler, sl. 13.c).



a) b) c)

d) e)

Sl. 13 Ostale korištene komponente: a) FTDI priključak b) Nosač baterije/ baterija 3V/ c) litij-ionska baterija 3.7V d) LilyPad LE dioda e) Konduktivni konac [32]

Za povezivanje komponenata u odjevni predmet koristi se konduktivni konac, sl. 12.e) čiji je električni otpor 8.5 Ωm^{-1} te je dodana LE dioda, sl. 13.d), dimenzije 5,5mm x 12,5mm debljine 0,8mm kao estetska komponenta čije svjetlo treperi u ritmu srčanog pulsa.

3.6 MJERENJE ELEKTRIČNOG SIGNALA POMOĆU OSCILOSKOPA

Za mjerenje električnog signala korišten je osciloskop koji se upotrebljava za očitavanje i promatranje promjene električnog signala tijekom određenog vremenskog perioda. Dakle, u ovisnosti o naponu i vremenu dobiva se oblik koji se neprekidno projicira prema kalibriranoj skali. Promatrani valni oblik može opisivati svojstva kao što su amplitude, frekvencije, vremenske intervale, distorzije i slično. Izvorni su, analogni osciloskopi, zahtijevali ručna mjerenja valnog oblika prema ugrađenim instrumentima. Danas, postoje i moderni digitalni osciloskopi koji mogu samostalno izračunati i prikazati svojstva valnog oblika. Osciloskopi se primjenjuju u medicini, inženjerstvu, automobilskoj i telekomunikacijskoj industriji. Opća namjena osciloskopa je održavanje elektroničke opreme i rad u laboratoriju. Osciloskopi s posebnom namjenom, kao što je slučaj u ovom radu, mogu i prikazivati valni oblik otkucaja srca kao elektrokardiogram.

Povezuje se USB kabelom pri čemu je sav prikaz i pohrana podataka omogućena na zaslonu računala.



Sl. 14 Osciloskop oznake PicoScope [34]

Klasični osciloskopi se dijele na četiri odjeljka. Zaslon, okomite i horizontalne kontrole te na kontrole pokretanja (engl. trigger) Zasloni mogu biti katodni ili u obliku LCD panela koji sadrže okomite i horizontalne linije tvoreći mrežu. Osim zaslona, postoje i tri osnovne kontrole, za izoštravanje, kontrola za pronalaženje zrake i kontrola za povećavanje i smanjivanje intenziteta

zrake. Dok analogni osciloskopi koriste različite napone, digitalni osciloskopi koriste binarne brojeve koji odgovaraju različitim uzorcima napona. Valovi su prikupljeni kao niz uzoraka koji se pohranjuju i skupljaju kako bi se sastavio valni oblik. Novije inačice digitalnih osciloskopa uključuju osciloskope za računala. To su osciloskopi koji se priključuju na računalo za obradu i prikaz podataka. Sastoje se od specijalizirane ploče za prikupljanje signala koja se s računalom mogu povezati. Pico Tehnology je britanski proizvođač visokopreciznih osciloskopa za računala sa samohodnom dijagnostikom. Svi Pico Tehnology modeli sadrže integrirani generator funkcija ili proizvoljni generator valnih oblika, pokretače te automatska mjerenja sa statistikom. Svaki PicoScope ima i softver koji omogućuje snimanje signala u stvarnom vremenu i snimanje valnih oblika. Omogućuje kontinuirani prikaz valnih oblika te olakšava i omogućava jednostavan pregled i usporedbu s prethodnim mjerenjima.

U ovom diplomskog radu, PicoScope je korišten kako bi se ispitala pouzdanost senzora, utjecaj i važnost napajanja te različite valne oblike dobivene od različitih ispitanika.

4. EKSPERIMENTALNI DIO I REZULTATI

Prema prethodno opisanom poglavlju, u eksperimentalno dijelu ovog rada će biti prikazano projektiranje elemenata pametne kape za praćenje signala srčanog pulsa.

4.1 PROGRAMIRANJE MIKROKONTROLERA PAMETNE KAPE ZA PRAĆENJE SIGNALA SRČANOG PULSA

Arduino, kao što je već spomenuto, je tip platforme otvorenog koda. Namijenjen je izradi različitih projekata i može se koristiti za programiranje bilo kakvih mikrokontrolera koji su u vezi s Arduinom. S obzirom da je Simblee posebna tvrtka usmjerena na proizvodnji Bluetooth uređaja te povezivanjem tih uređaja s mobitelima, tabletima i računalima, potrebno je Arduino modificirati kako bi programiranje Simblee-ja bilo moguće.

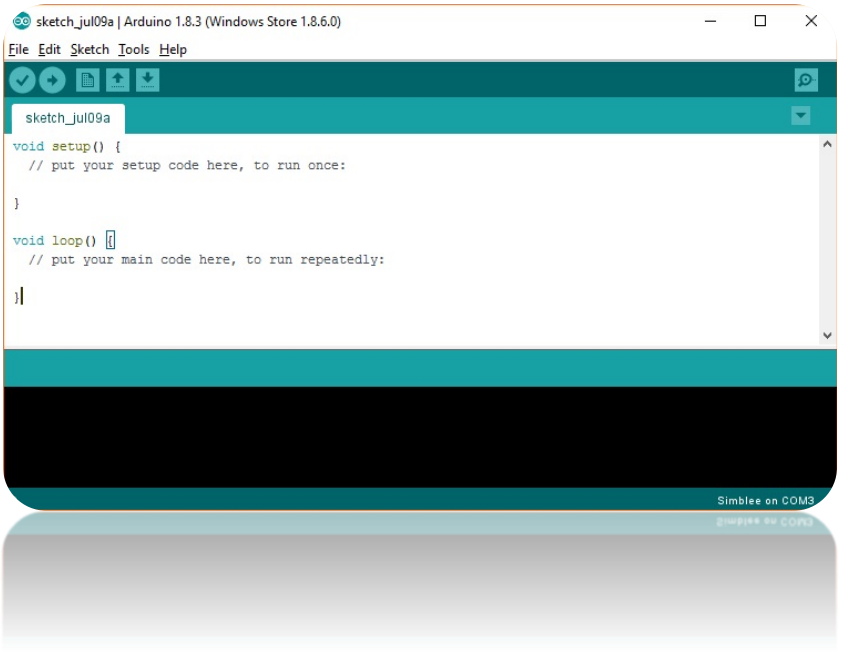
Nakon što se program Arduino IDE 1.8.3 uspješno instalira na osobno računalo, sl. 15, slijedi instalacija Simblee baza podataka za programiranje Simblee mikrokotrolera te instalacija upravljačkih programa. Upravljački programi su potrebni kako bi Lilypad Simblee BLE povezali FTDI priključkom na računalu.

Proces programiranja koda je izuzetno olakšan zahvaljujući jednostavnom i otvorenom pristupu Interneta. Kao što je već objašnjeno Arduino je tip platforme otvorenog koda što znači da su mnogi projekti i radovi dostupni i slobodno se mogu koristiti i izmjenjivati.

Internetska stranica pod nazivom GitHub [35] je razvojna platforma čiji se sustav temelji na radovima i projektima znanstvenika, studenata, kreativaca, umjetnika i slično. GitHub je sustav koji omogućuje stvaranje programa, pregled i upravljanje istih te je stvoren s ciljem kako bi surađivali, učili i unapređivali jedni od drugih.

U procesu istraživanja kodova i načina programiranja, za potrebne ovog diplomskog rada, proučeni su vrlo slični projekti gdje je, upravo zahvaljujući otvorenom pristupu, omogućeno korištenje i prilagođavanje programa.

Program koji je korišten u ovom diplomskom radu je Fitness bracer. S obzirom da se ovaj diplomski rad služi jednim senzorom, i to senzorom Fitness bracer [36] za mjerenje otkucaja srčanog pulsa, bilo je potrebno prilagoditi odnosno pojednostaviti pronađeni program. U suradnji sa studentom s Fakulteta elektrotehnike i računalstva prilagodba programa učinjena je tako da smo dijelove programa koji nisu potrebni, u računalnom jeziku, „zakomentirali“ oznakom odnosno simbolom dvije kose crte //. Na taj način su se zanemarili dijelovi koda koji za svrhu ovog rada nisu potrebni te se oni neće prikazati prilikom prebacivanja koda na Lilypad Simblee BLE.



Sl. 15 Sučelje Arduino-a na kojem je instaliran Simblee baza podataka

Instalacija programa preuzeta je sa službenih stranica Arduina i Simbleeja te je za instalaciju upravljačkih programa korištena službena uputa na njihovim stranicama.

Simblee aplikacija je također izmijenjena na način da je pojednostavljen glavni prozor aplikacije. Na vrlo jednostavan način se izmijenila aplikacija, informacije koje su bitne te izmijenila boja i veličina teksta.

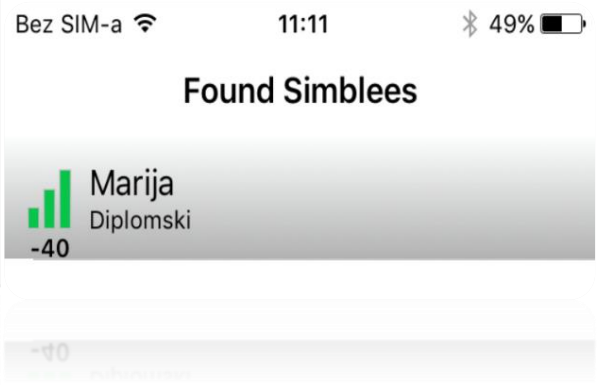
Pomoću FTDI priključka spojen je Lilypad Simblee BLE s računalom i prebačen je kod na mikrokontroler. Kao potvrda uspješnog učitavanja koda Arduino šalje obavijesti porukom „*success*“. U prilogu diplomskog rada, u tiskanom formatu, nalazi se dio programa koji se odnosi na programiranje senzora za očitavanje pulsa. Cijeli program priložen je u elektroničkom formatu rada.

Nakon uspješnog programiranja uslijedila je provjera rada programa, senzora te aplikacije.

Sučelje SimbleeForMobile koji je korišten unutar programa omogućuje spajanje Simblee mikrokontorlera s prikladnom Simblee aplikacijom, sl. 16. SimbleeForMobile je već postojeća aplikacija koja se može naći na Google Playu ili Apple Storeu. To je aplikacija isprogramirana od strane Simblee tvrtke koja je u stanju identificirati Niskoenergetske Bluetooth uređaje. Kada se spoje, korisniku pruža mogućnost odabira željenog Simblee uređaja. Pametni telefon u ovom slučaju pruža sučelje za interakciju sa Simblee mikrokontolerom. Prilikom provjere rada

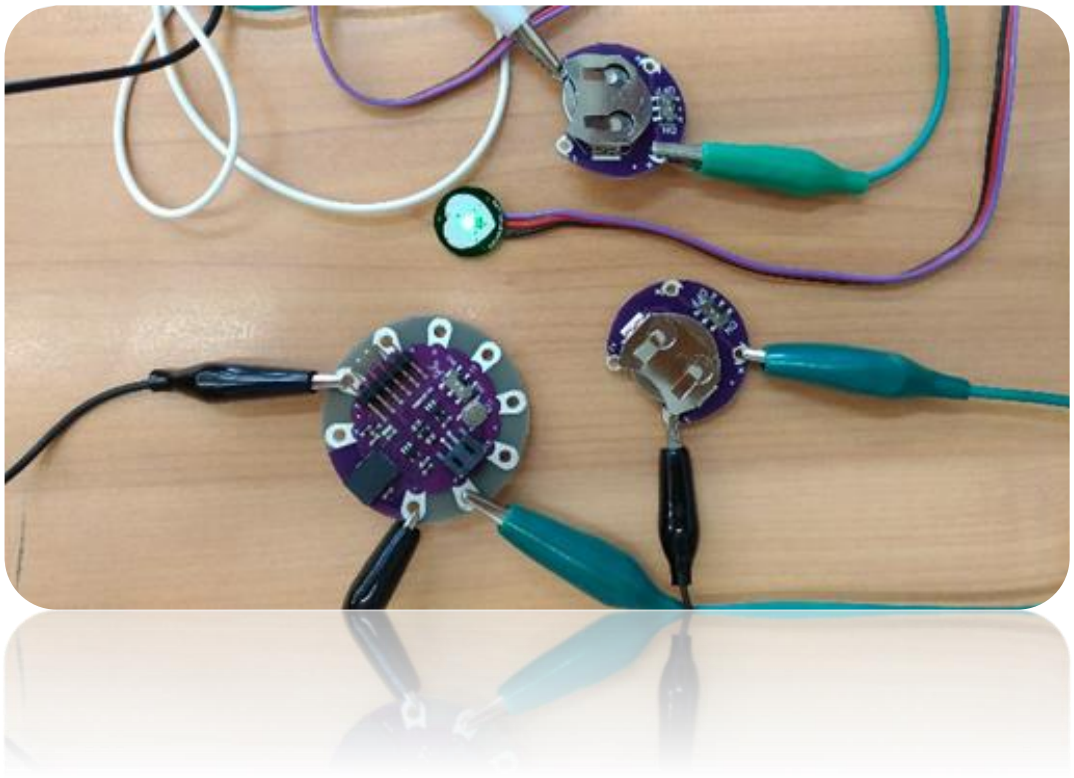
programa sa Simblee aplikacijom, naišlo se na zapreku što se tiče odabira pametnog telefona odnosno operativnog sistema. Uspješno preuzimanje Simblee aplikacije ostvaren je pomoću IOS operativnog sistema, sl. 17, dok je za Andriod još uvijek nekompatibilan.





Sl. 16 Simblee aplikacija

Sl. 17 Simblee aplikacija povezana sa mikrokontrolerom Bluetooth tehnologijom

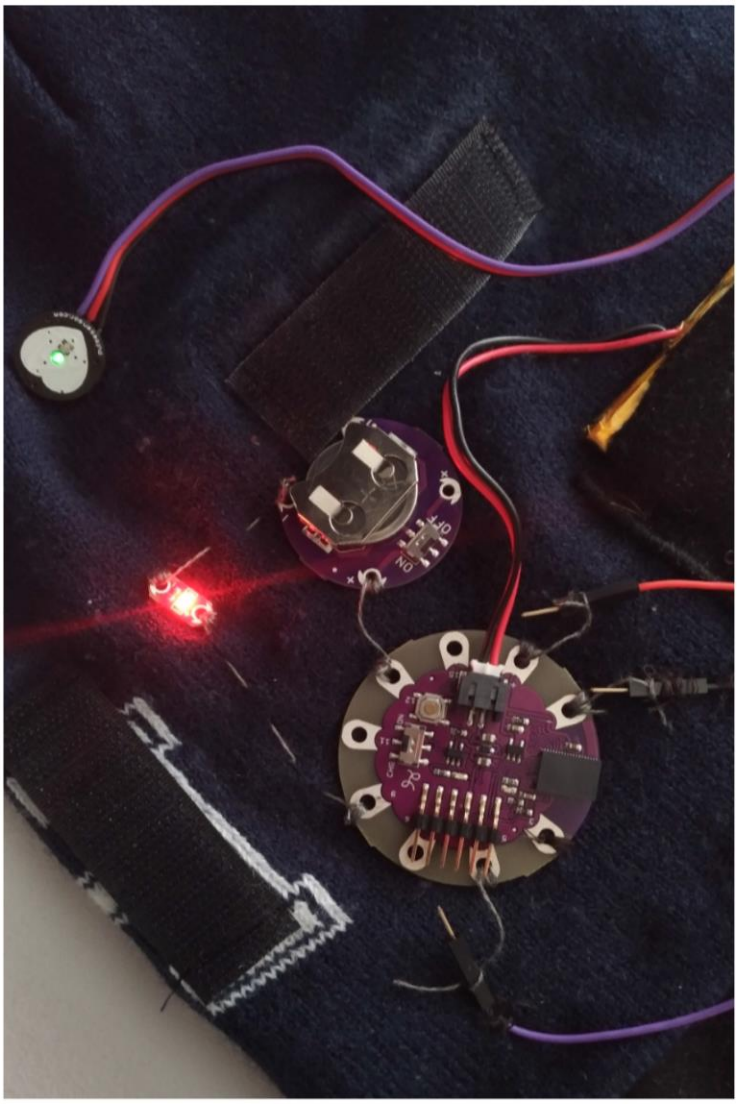


Sl. 18 Povezivanje Lilypad Simblee BLE, senzor i baterije pomoću tzv. „klema krokodilki“

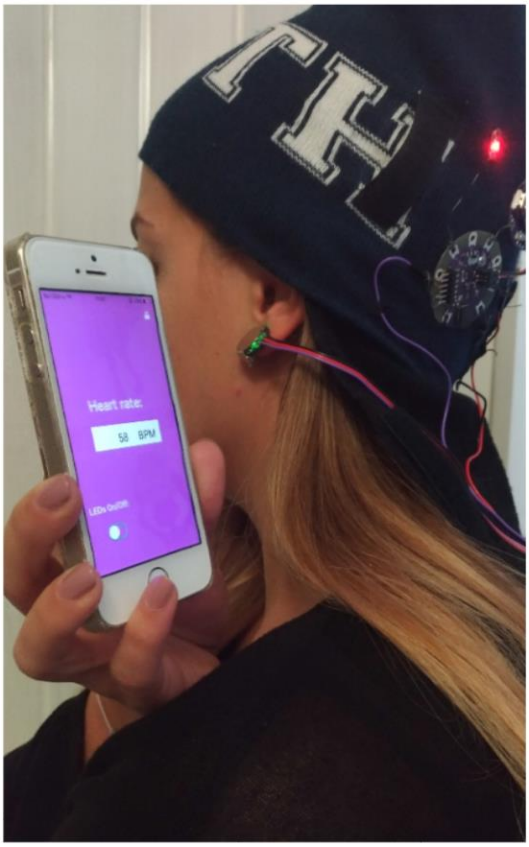
Za uspješno povezivanje senzora i Lilypad Simblee BLE mikrokontrolera korištene su tzv. „kleme krokodilke“ kako bi se prenio električni signal sa senzora na mikrokontroler. S obzirom da je ispitivanje svih komponenta bilo u početnoj fazi, sl. 18, ovo je najučinkovitiji način provjere prije samog spajanja svih dijelova i komponenata u pametnu kapu.

4.2 PROJEKTIRANJE PAMETNE KAPE ZA PRAĆENJE SIGNALA SRČANOG PULSA

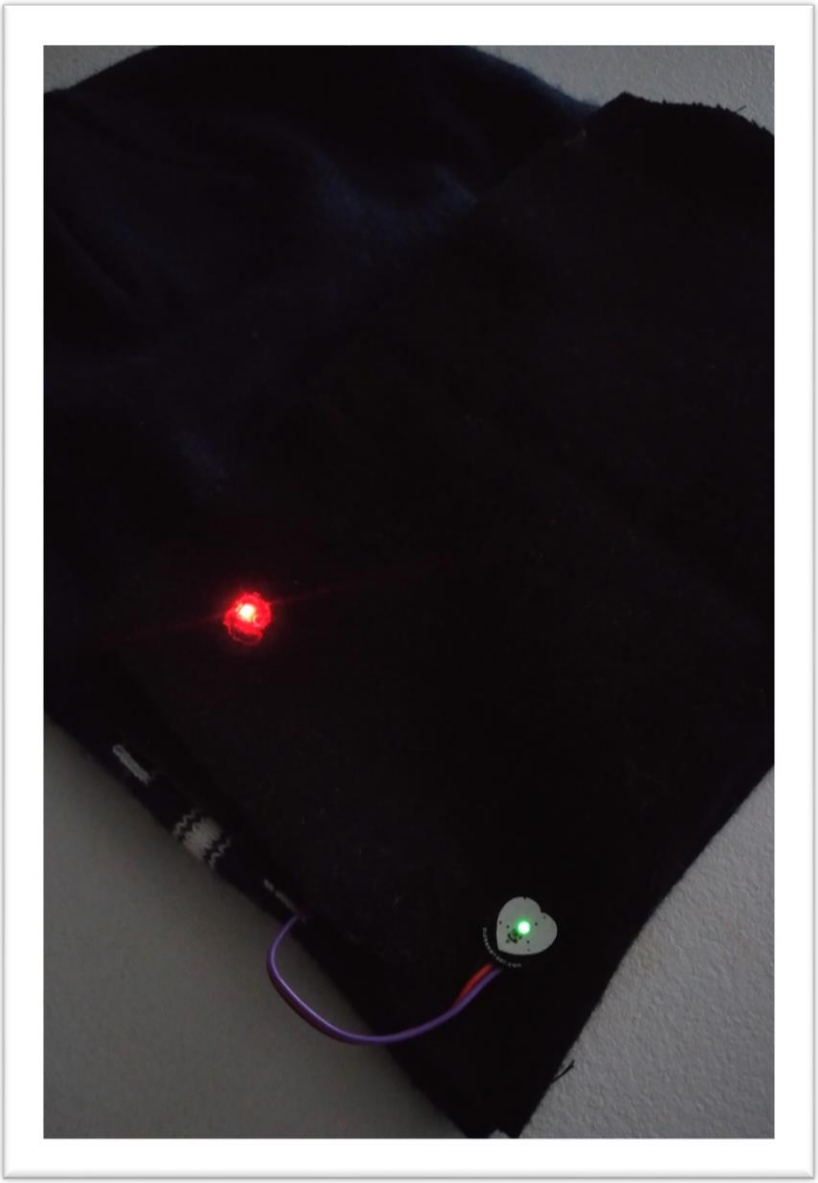
Nakon provjere rada svih komponenti pristupilo se integraciji svih elemenata u pametnu kapu za praćenje signala srčanog pulsa.



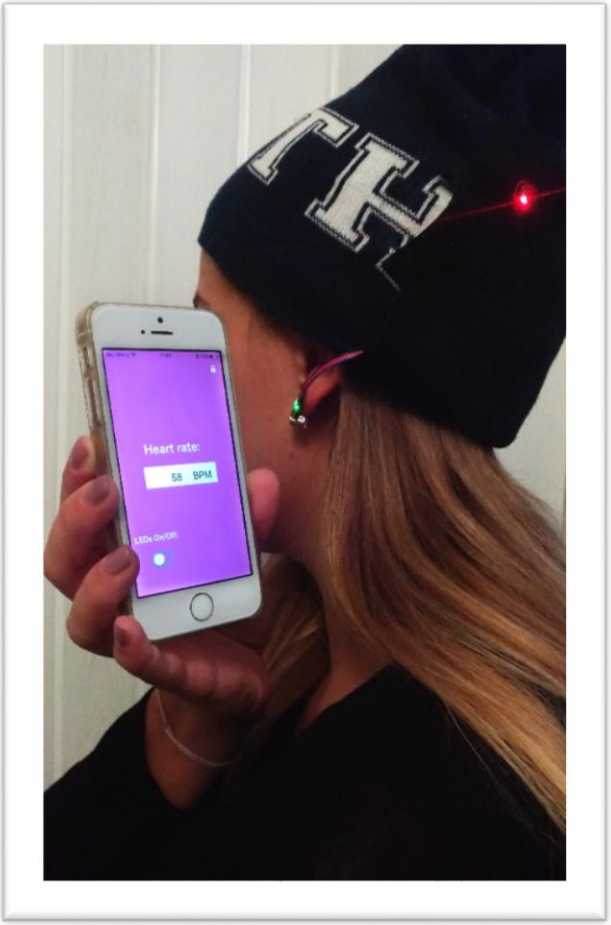
a)



b)



c)



d)

Sl. 19 Pametna kapa za praćenje signala srčanog pulsa s integriranim elementima

Na sl. 19.a) je prikazana unutrašnjost pametne kape s integriranim elementima za praćenje signala srčanog pulsa, a na sl. 19.b) je prikazana osoba s pametnom kapom, sa svim integriranim elementima, kako prima signale srčanog pulsa na mobilni aparat. Sl. 19.c) vanjski izgled pametne kape. 19.d) je prikazan vanjski izgled pametne kape na nositelju koji također signale srčanog pulsa prima na mobilni aparat

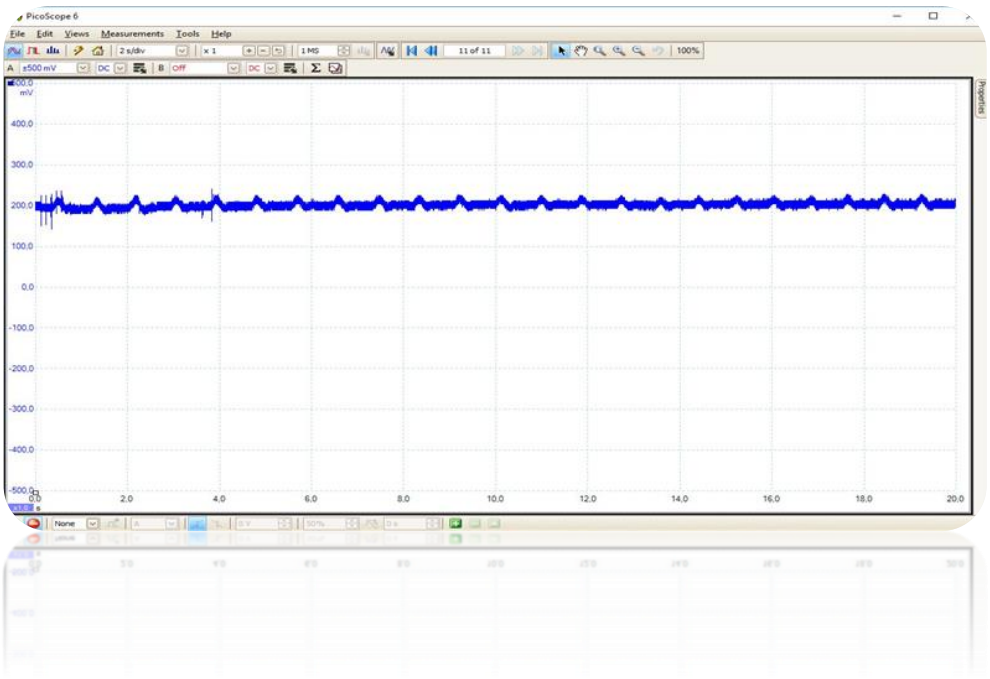
4.3 ISPITIVANJE SENZORA OSCILOSKOPOM

Pri mjerenju otkucaja srca te prateći rezultate na aplikaciji, u pojedinim trenutcima, dobiveni su netočni i nepouzdani podaci u odnosu na standardni broj otkucaja srca po minuti, koji kod odrasle osobe odgovara brojci između 60 do 100. Problem nastaje zbog nepreciznog broja otkucaja srca i velikih odstupanja. Mogući razlozi zbog kojih dolazi do nepreciznih mjerenja mogu biti kvaliteta senzora, način odnosno jačina napajanja te položaj senzora kao i tlačna sila koja djeluje na senzor pri dodiru s kožom. Također, s obzirom da se radi o optičkom senzoru, svaka svjetlosna promjena u intenzitetu svjetla se očitava, što znači da je potrebno jednakomjerno očitavanje duži period kako bi se broj otkucaja srca po minuti mogao očitati.

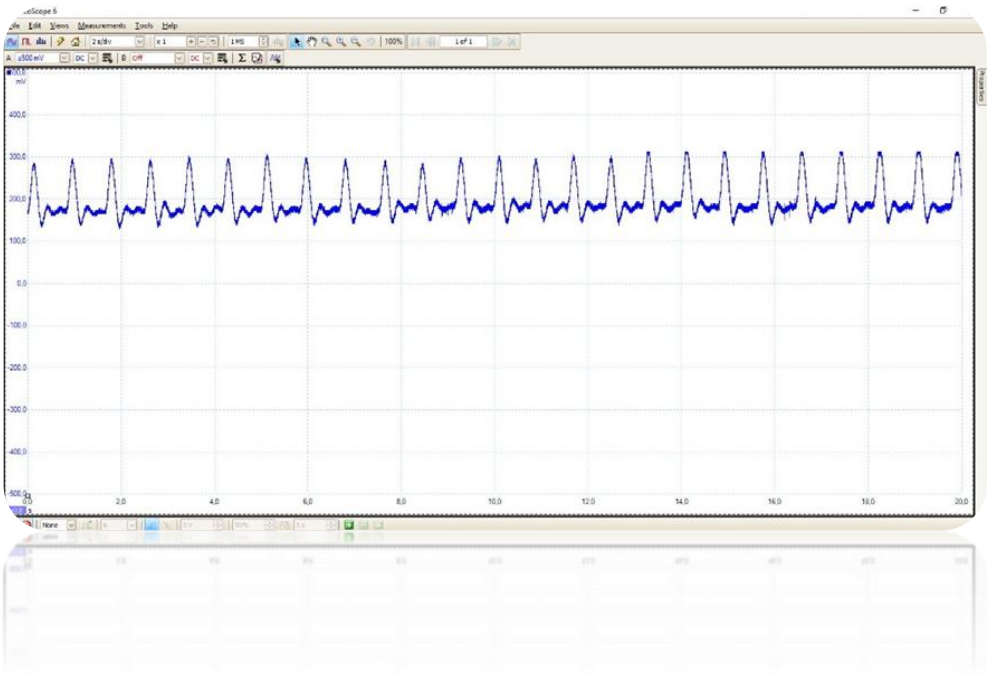
S obzirom da je senzor koji se koristi u ovom radu namijenjen *plug and play* svrsi racionalno gledajući ne mogu se u potpunost očekivati rezultati kako bi oni u uvjetima stvarnog ispitivanja trebali biti. Također je primijećeno da iznos napajanja bitno utječe na preciznost rada i prikaz podataka. Prilikom mjerenja otkucaja srca kada mikrokontroler Lilypad Simblee BLE dobiva napajanje preko USB kabela, priključen na računalu, dobivaju se precizniji rezultati. Količina napajanja preko računala je 5 Volta dok je preko baterije ograničena na samo 3 Volta.

Ispitivanjem senzora na osciloskopu dokazano je da napajanje i vrsta napajanja utječe na jačinu signala koji senzor dobiva ali i na prijenos tih podataka. Kada je senzor spojen na mikrokontroler čije je napajanje dobiveno preko USB kabela dobiven je jači prikaz pulsa nego kada je napajanje prisutno pomoću baterije od 3 Volta.

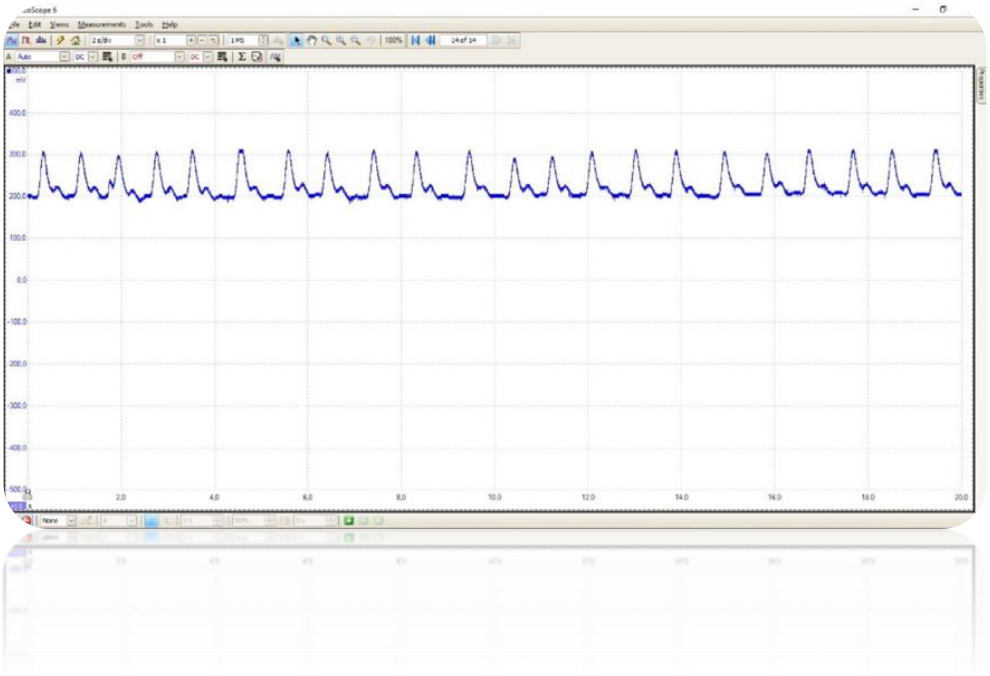
Iz sljedećih grafičkih prikaza (sl. 20-22) vidljiv je utjecaj napajanja na senzor. Sl. 20 prikazuje rad senzora kada se senzor i mikrokontroler napajaju koristeći samo bateriju. Iz sl. 21 i 22 vidljiv je rad senzora kada je mikrokontroler uključen na USB kabel čije napajanje je postignuto koristeći snagu računala. Amplituda signala sa sl. 20 znatno je manja nego ona sa sl. 21 i 22.



Sl. 20 Ispitivanje senzora osciloskopom koristeći bateriju od 3V



Sl. 21 Prikaz mjerenja srčanog pulsa starije osobe na osciloskopu



Sl. 22 Prikaz mjerenja srčanog pulsa mlađe osobe na osciloskopu

Na sl. 21 prikazano je mjerenja srčanog pulsa pomoću senzora smještenog u kapi, kod mlađe osobe, a na sl. 22 kod starije osobe.

5. RASPRAVA

Senzor temelji svoj rad na mjerenju refleksije prokrvljenog i neprokrvljenog tkiva kože čovjeka. U trenutku kada srce ubrizgava krv u žile prokrvljenost tkiva je veća, a kad miruje prokrvljenost je manja. Senzor posjeduje emitirajuću LE diodu koja obasjava tkivo i prijemnu LE diodu koja mjeri refleksiju. Na temelju razlika u refleksiji tkiva može se određivati puls srca.

Svrha ovog diplomskog rada temelji se na projektiranju prototipa pametna odjeće koji je u stanju mjeriti ali i motriti stanje otkucaja srčanog pulsa. Istraživanje se temelji na odjeći koja bi pomogla bolesnicima i rekonvalescentima, u onim trenutcima, kada je potrebno dugoročno bilježenje rada srca ili kada se provodi određena terapija. Temeljna ideja je izraditi prototip odjevnog predmeta koji je u skladu s brzim razvojem tehnologije kojom se svakodnevno susrećemo, omogućavajući jednostavan i interaktivan način praćenja rada srca svakog individualnog nositelja odjevnog predmeta. Razmišljajući o spoju tehnologije i odijevanja rodila se ideja za stvaranje odjevnog predmeta koji će mnogima omogućiti praćenje rada srca. Podaci otkucaja srca će se mjeriti pomoću senzora, a na osobnom pametnom telefonu, putem Bluetooth-a i prikladne mobilne aplikacije, bit će vidljivi podaci.

Najčešći znak veće aktivnosti čovjeka je povećani rad srca koji se mjeri u otkucaju po minuti (engl. beats per minute – BPM). Povećanje broja otkucaja srca tijekom vježbanja ili kretanja pomaže pri održavanju homeostaze našeg organizma, no međutim, povećani ili smanjeni rad srca može biti vrlo opasan. Uređaji koji se u medicini koriste za motrenje otkucaja srca služe za dijagnosticiranje i nadziranje zdravlja.

Nakon što je donesena odluka o području mjerenja uslijedila je analiza potrebnih senzora, mikrokontrolera, procesora i programa bez kojeg takva odjeća ne bi sadržala atribute pametne odjeće.

Nakon uspješnog programiranja, provjere programa, aplikacije i rada senzora pristupilo se izradi prototipa pametne odjeće.

Kao temeljni odjevni predmet izabrana je zimska kapa. Ugradnja Lilypad Simblee BLE mikrokontrolera, senzora, LE diode i baterija postignuto je koristeći konduktivni konac kako bi se omogućio protok struje kroz elemente. Kao krajnji rezultat dobiven je odjevni predmet pomoću kojeg se nesmetano može pratiti otkucaj srčanog pulsa. Senzor se pričvršćuje za ušnu resicu te se na taj način dobivaju podaci o otkucaju srca po minuti. Senzor na ušnoj resici prati varijacije u svjetlosnom intenzitetu koji nastaje zbog promjene volumena prouzrokovan prolaskom krvi kroz područje mjerenja. Koristeći aplikaciju i Bluetooth tehnologiju ti podaci vidljivi su na zaslonu pametnog telefona. LE dioda svijetli crvenom bojom kao indikator da je odjevni predmet uključen odnosno da je u funkciji. Kao što je opisano u prethodnom poglavlju,

senzor je namijenjen *plug and play* svrsi što znači da je upitna ispravnost podataka koju dobivamo na zaslonu pametnog telefona. Ujedno i količina napajanja utječe na preciznost podataka te način spajanja komponenata. Projektiranje pametne kape za praćenje srčanog pulsa ostvarena je s ciljem da se prikaže naprednost i spoj tehnologije u odjevnoj industriji kao i multidisciplirarnost pametne odjeće. Buduća istraživanja temeljit će se na integraciji manjih senzora i komponenata te na dizajnu koji će omogućavati manju uočljivost i fleksibilnost prilikom nošenja.

Koristeći osciloskop ispitan je utjecaj napajanja na senzor. Senzor temelji svoj rad na mjerenju refleksije prokrvljenog i neprokrvljenog tkiva kože čovjeka. U trenutku kada srce ubrizgava krv u žile prokrvljenost tkiva je veća, a kad miruje prokrvljenost je manja. Senzor posjeduje emitirajuću LE diodu koja obasjava tkivo i prijemnu LE diodu koja mjeri refleksiju. Na temelju razlika u refleksiji tkiva može se određivati puls srca.

Iz oscilograma prikazanih na sl. 20, 21 i 22 je vidljivo, tijekom istraživanja, da je najbolje izabrati razinu signala iz senzora na razini od 0,25 V (250 mV) što je moguće koristeći programsku podršku osciloskopa koji se priključuje na računalo.

6. ZAKLJUČAK

U ovom radu je prikazano projektiranje elemenata pametne odjeće za praćenje srčanog pulsa na primjeru zimske kape. Signal se prati, koristeći biosenzor, LilyPad elemente za ugradnju u odjeću međusobno povezanih konduktivnim koncem, bežičnu komunikaciju te programsku podršku, na mobilnom aparatu (iPhone-u). Za potrebe programiranja korištena je tzv. „open source“ programi na Arduino projektiranje elemenata pametne odjeće za praćenje signala srčanog pulsa platformi, koja je namijenjena elektroničarima, dizajnerima i svima ostalima koji žele stvarati interaktivne proizvode.

Ovaj diplomski rad je nastavak dugogodišnjeg rada znanstvenika na Sveučilištu u Zagrebu, tekstilno-tehnološkom fakultetu na razvoju inteligentne i pametne odjeće, gdje su odnedavno uključeni i studenti, kako bi razvili svoju kreativnost i inovativnost.

Pametna odjeća integrira dizajn, funkcionalnost odjeće i prenosivih tehnologija te je stoga razvoj tehnologije i multidisciplinarnost razvojnih timova unatrag nekoliko desetljeća dovelo je do proizvodnje i uporabe minijaturiziranih niskoenergetskih senzora i elektroničkih komponenata te njihove integracije u odjeću. Projektiranje pametne kape za praćenje signala srčanog pulsa, u sklopu ovog diplomskog rada je rezultiralo funkcionalnim prototipom pametnog odjevnog predmeta koji se može usavršavanjem programske podrške i daljnjom minijaturizacijom elemenata, uspješno koristiti za praćenje i motrenje zdravstvenog stanja različitih skupina ljudi: od djece do strije populacije, od bolesnika i rekreativaca do profesionalnih sportaša i vojnika.

Može se zaključiti da će pametna odjeća u bliskoj budućnosti postati naša svakodnevica kao što je to slučaj s raznim nosivim uređajima kao i pametnim telefonima koji su postali neizostavni dio našeg života. Odjevna industrija počela je razvijati pametnu odjeću za koju se bez sumnje može zaključiti da je budućnost odijevanja povezana digitalnim svijetom.

7. REFERENCE

[1] S. Frišt Rogale, D. Rogale, G. NIkolić, Z. Dragčević, Inteligentna odjeća, Sveučilište u Zagrebu Tekstilno-tehnološki fakultet, ISBN: 978-953-7105-52-5, 2014..

[2] S. Mann, »Wearable Computing: A First Step Toward Personal Imaging,« *Computer*, svez. 30, br. 2, pp. 25-32, 1997.

[3] C. R. Rachana, »An insight into the world of Wearable Computing,« *2014*, svez. 3, br. 3, pp. 491-495, Engineering and Technology Research (IJSETR).

[4] E. El-Khatib, »Antimicrobial and Self-cleaning Textiles using Nanotechnology,« *Research Journal of Textile and Apparel* , svez. 16, br. 3, pp. 156-174, 2012.

[5] U. K. Chandhok, »Smart Clothing: Technological Innovations,« 5. 1. 2016. :https://www.linkedin.com/pulse/smart-clothing-technological-innovations-udit-karan-chandhok. [Pristupljeno 1 9 2017].

[6] D. B. J. McCann, u *Smart Clothes and Wearable Technology* , ISBN 9781845693572, Woodhead Publishing, 2009, pp. 183-215.

[7] A. M. N. T. J. Farringdon, »Wearable sensor badge and sensor jacket for context awareness,« u *Proceeding ISWC '99 Proceedings of the 3rd IEEE International Symposium on Wearable Computers* .

[8] ..., »Induktivni, kapacitivni, optički i ultrazvučni senzori prisutnosti. Cilj vježbe: Upoznavanje sa senzorima prisutnosti i snimanje njihovih karakteristika. UVOD.,« :www.ieee.hr/_download/repository/PM_vjezba_1.pdf. [Pristupljeno 1. 9. 2017].

[9] M. Tamsin, »Wearable Biosensor Technologies,« *Internation Journal of Innovation and Scientific Research, ISSN 2351-8014* , svez. 13, br. 2, pp. 697-703, 2015.

[10] ..., »Kemijski senzori,« :https://www.fer.hr/_download/repository/MP11_BioKemijskiSenzori.pdf. [Pristupljeno 5. 9. 2017].

[11] V. R. G. Malčić, »Mikrokontrolerski sustavi,« Tehničko veleučilište Zagreb, :nastava.tvz.hr/gmalcic/ERRO_knjiga_TVZ.pdf. [Pristupljeno 25 8 2017].

[12] P. Zenzerović, Arduino kroz jednostavne primjere,, Zagreb: Hrvatska zajednica tehničke kulture, 2014.

[13] W. C. Q. Wang, »Literature review on wearable systems in upper extremity rehabilitation,« u *Biomedical and Health Informatics, DOI: 10.1109/BHI.2014.6864424* , Valencia, Spain , 2014.

[14] H. P. P. B. L. C. M. R. S. Patel, »A review od wearable sensors and systems with application in rehabilitation,« *Journal of NeuroEngineering and Rehabilitation,*, pp. DOI: 10.1186/1743-0003-9-21, 2012.

[15] M. T. M. M. J. D. S, »Wearable Sensors for Remote Health Monitoring,,« svez. DOI:10.3390/s17010130, 2017..

[16] K. C. N. C. M. Suh, » Critical Review on Smart Clothing Product Development,« *ournal of Textile and Apparel, Technology and Management*, svez. 6, br. 4, 2010.

[17] A. P. Hum, »Fabric area network – a new wireless communications infrastructure to enable ubiquitous networking and sensing on intelligent clothing,« *Computer Networks, DOI: 10.1016/S1389-1286(00)00185-7*, svez. 35, br. 4, pp. 391-399, 2001.

[18] N. Pađina, »Body Area Network,« :http://spvp.zesoi.fer.hr/seminari/2007/seminari/NikaParadina_BodyAreaNetwork.pdf. [Pristupljeno 20. 8. 2017].

[19] .... :https://www.emaze.com/@AWLFCCFO/Ra%C4%8Dunarske-mre%C5%BEe. [Pristupljeno 21. 8. 2017].

[20] .... :http://download.sony-europe.com/pub/manuals/html/Z003907111/HR/contents/TP0001034966.html. [Pristupljeno 2 9 2017].

[21] .... :http://www.techionix.com/wp-content/uploads/2016/03/117.jpg. [Pristupljeno 20. 8. 2017].

[22] S. Firšt Rogale, D. Rogale, »Obuća i odjeća kao visokotehnologijski predmet u funkciji izvora električne energije,« u *Antropometrija i normizacija modne odjeće*, Zagreb, Sveučilišna tiskar, 2015, pp. 23-34.

[23] .... :https://www.google.hr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=0ahUKEwiw8brL0L7WAhULYlAKHaL6AXQQjBwIBA&url=http%3A%2F%2Fecofriend.com%2Fwp-content%2Fuploads%2F2012%2F07%2Fsolar_panels_in_jacket.jpg&psig=AFQjCNFoVXt3_zW7BwIkB6yRlJV9TSjgKQ&ust=1506370. [Pristupljeno 20. 8. 2017].

[24] .... :https://www.geek.com/wp-content/uploads/2011/08/energy_harvest_shoe_02.jpg. [Pristupljeno 20. 8. 2017].

[25] ..., »Life Shirt - Vivosense,« :https://www.vivosense.com/portfolio/lifeshirt/. [Pristupljeno 20 8 2017].

[26] .... :http://fatknowledge.blogspot.hr/2005/03/. [Pristupljeno 24. 9. 2017].

[27] :http://slideplayer.com/slide/4215175/. [Pristupljeno 20. 8. 2017].

[28] J. Dutcher, » Massimo Banzi- How Arduino is Open-Sourcing Imagination,« :https://datascience.berkeley.edu. [Pristupljeno 20. 8. 2017].

[29] »Arduino,« :https://www.arduino.cc/. [Pristupljeno 20. 8. 2017].

[30] »Simblee,« :https://www.simblee.com/. [Pristupljeno 20. 8. 2017].

[31] »LilyPad Simblee BLE Board,« :https://www.sparkfun.com/products/13633. [Pristupljeno 2. 9. 2017].

[32] ..., svez. 20, br. https://www.sparkfun.com/, p. 7, 2017.

[33] :https://pulsesensor.com/pages/pulse-sensor-amped-arduino-v1dot1. [Pristupljeno 8.8. 2017].

[34] :https://www.picotech.com/. [Pristupljeno 25.8.2017].

[35] :https://github.com. [Pristupljeno 3.7. 2017].

[36] :https://github.com/sparkfun/Simblee-LilyPad-Fitness-Bracer. [Pristupljeno 7.7. 2017].

[37] ..., »Renesas Solutions for Wireless Sensor Networks—Part 2: Body Area Networks,« :https://www.renesas.com/en-us/about/web-magazine/edge/solution/08-body-area-networks.html . [Pristupljeno 20. 6. 2017].

Prilog:

```
//Simblee Lilypad Fitness Bracer Code
//Sarah Al-Mutlaq
//March 8th 2016
//Written in Arduino 1.6.7
//SparkFun Electronics

// To use the SimbleeForMobile library, you must include this file at the top
// of your sketch. **DO NOT** include the SimbleeBLE.h file, as it will cause
// the library to silently break.
#include<SimbleeForMobile.h>

//Define pins:

const uint32_t LEDPIN = 9;
const int pulsePin = 5;
const int buttonPin = 15;
const int vibePin = 12;
const int vibePin2 = 11;
const int accXPin = 6;
const int accYPin = 3;
//const int tempPin = 4;


//Floats for calculating accelorometer changes to get steps:
float threshhold=300.0;

float xval[100]={0};
float yval[100]={0};

float xavg = 173.10;
float yavg = 0.00;

int flag=0;


//Names and identifiers for the drawn stuff on the app:
uint8_t rectangle1ID;
uint8_t rectangle2ID;
uint8_t button1ID;
uint8_t addCalBtnID;
uint8_t button3ID;
uint8_t LEDswitchID;
uint8_t text1ID;
uint8_t text2ID;
uint8_t text3ID;
uint8_t text3unitID;
uint8_t text4ID;
uint8_t text5ID;
```

```
uint8_t text6ID;
uint8_t text7ID;
uint8_t lastActTextID;
uint8_t activeUnitID;
uint8_t text9ID;
uint8_t text10ID;
uint8_t text10UnitID;
uint8_t text11ID;
uint8_t text12ID;
uint8_t text12UnitID;
uint8_t text13ID;
uint8_t calTextFldID;
uint8_t text15ID;
uint8_t text16ID;
uint8_t text16UnitID;
uint8_t segment1ID;
uint8_t InputTextFldID;

//Place holders for values we get/calculate:
int16_t STEPS = 0;
float steps = STEPS;
int16_t STRIPCHOICE = 0;
const char* stripChoiceUnit = " ";
int16_t LASTACTIVITY = 0;
const char* activityUnit = "mins";
int16_t CURRENTTEMP = 0;
const char* tempUnit = "F";
int16_t LASTHEARTRATE = 95;
const char* heartUnit = "bpm";
int16_t Add = 0;
int16_t caloriesTotal = 0;
int16_t currentScreen = -1;
int16_t stripSelection = 4;


//Some variables to keep track of time and states (since you can't have blocking code):
int buttonState = HIGH;
long lastStepDebounceTime = 0;
long stepDebounceDelay = 30;
long lastTimer = 0;
long newTimer = 0;

//Values for the red, green, and blue parts of the RGB LED:
int redVal;
int greenVal;
int blueVal;

//Variable for your daily step goal (for the LED indicator):
float stepGoal = 10000;
```

```
float percentSteps = (steps/stepGoal);

//Indicator for whether the LED should be on or off (1 = on):
int leds = 1;


//Variables to get and calculate heart rate from the pulse sensor:
volatile int BPM;                   // used to hold the pulse rate
volatile int Signal;                // holds the incoming raw data
volatile int IBI = 600;             // holds the time between beats, the Inter-Beat Interval
volatile boolean Pulse = false;     // true when pulse wave is high, false when it's low
volatile boolean QS = false;        // becomes true when Arduoino finds a beat.

volatile int rate[10];                    // used to hold last ten IBI values
volatile unsigned long sampleCounter = 0;          // used to determine pulse timing
volatile unsigned long lastBeatTime = 0;           // used to find the inter beat interval
volatile int P =512;                      // used to find peak in pulse wave
volatile int T = 512;                     // used to find trough in pulse wave
volatile int thresh = 512;                // used to find instant moment of heart beat
volatile int amp = 100;                   // used to hold amplitude of pulse waveform
volatile boolean firstBeat = true;        // used to seed rate array so we startup with reasonable BPM
volatile boolean secondBeat = true;       // used to seed rate array so we startup with reasonable BPM

//Setup function (only runs once at the start):
void setup()
{
  //Define our pins as inputs of outputs, and set ones that are outputs:
      pinMode(LEDPIN, OUTPUT);
      pinMode(buttonPin, INPUT_PULLUP);
      pinMode(pulsePin, INPUT);
      pinMode(vibePin, OUTPUT);
      pinMode(vibePin2, OUTPUT);
      digitalWrite(vibePin, LOW);
      digitalWrite(vibePin2, LOW);
      pinMode(accXPin, INPUT);
      pinMode(accYPin, INPUT);

//      pinMode(tempPin, INPUT); // if I had more anologe pins I would add a temp sensor

      Serial.begin(115200); //to be able to use the arduino serial monitor terminal to debug
      Serial.println("setup");

//      calibrate(); //Function you can call to set the average X and Y from the accelorometer
                  //so that it is more accurate about your steps (uncomment function to run).


      // advertisementData shows up in the app as a line under deviceName. Note
      // that the length of these two fields combined must be less than 16
      // characters!
```

```
        SimbleeForMobile.advertisementData = "FitnessTracker";
        SimbleeForMobile.deviceName = "5";
        // txPowerLevel can be any multiple of 4 between -20 and +4, inclusive. The
        // default value is +4; at -20 range is only a few feet.
        SimbleeForMobile.txPowerLevel = -4;

        // This must be called *after* you've set up the variables above, as those
        // variables are only written during this function and changing them later
        // won't actually propagate the settings to the device.
        SimbleeForMobile.begin();
}


//Function that runs continuously looping:
void loop()
{
  Serial.println("loop");
  //Check the button pin to see if it has been switched:
  int buttonRead = digitalRead(buttonPin);

  Serial.println(percentSteps); //for debugging

//These if functions set the LED different colors based on how close you are to your step goal:
   if ((percentSteps <= 0.25) && (leds == 1))
       {
          redVal = 0xff;
          greenVal = 0x00;
          blueVal = 0x00;
       }
   else if ((percentSteps <= 0.5) && (leds == 1))
       {
          redVal = 0xff;
          greenVal = 0xb3;
          blueVal = 0x00;
       }
   else if ((percentSteps <= 0.75) && (leds == 1))
       {
          redVal = 0xff;
          greenVal = 0xff;
          blueVal = 0x00;
       }
   else if ((percentSteps <= 1) && (leds == 1))
       {
          redVal = 0x00;
          greenVal = 0xff;
          blueVal = 0x00;
       }
   else if ((percentSteps > 1) && (leds == 1))
       {
```

```
        redVal = 0x00;
        greenVal = 0xff;
        blueVal = 0xff;
      }
   if (leds == 0)
      {
        redVal = 0x00;
        greenVal = 0x00;
        blueVal = 0x00;
      }

    RGB_Show(redVal,greenVal,blueVal, LEDPIN);


//This a timer so that it can keep track of the last time you were active:
      static uint32_t timer = 0;
      long timerMins = 0;
      timer = millis()/1000;
      timerMins = timer/60;
      if (SimbleeForMobile.updatable)
      {
       newTimer = timerMins - lastTimer;
       if(currentScreen == 2){
         SimbleeForMobile.updateValue(lastActTextID, newTimer);
       }
       if(currentScreen == 1){
          LASTACTIVITY = newTimer;
          if (stripSelection == 0){
            STRIPCHOICE = LASTACTIVITY;
          SimbleeForMobile.updateValue(text3ID, STRIPCHOICE);
          }
       }
      }

//This is the code that caluclates when a step has been taken:
  int accelerationX = analogRead(accXPin);
  int accelerationY = analogRead(accYPin);

  int acc=0;
  float totvect[100]={0};
  float totave[100]={0};
  float xaccl[100]={0};
  float yaccl[100]={0};

    for (int i=0;i<100;i++)
    {
    xaccl[i]=float(analogRead(accXPin));
    yaccl[i]=float(analogRead(accYPin));
```

```
    totvect[i] = sqrt(((xaccl[i]-xavg)* (xaccl[i]-xavg))+ ((yaccl[i] - yavg)*(yaccl[i] - yavg)));
    totave[i] = (totvect[i] + totvect[i-1]) / 2 ;
    if ((millis() - lastStepDebounceTime) > stepDebounceDelay)
    {
          if (totave[i]>threshhold && flag==0)
          {
            STEPS = STEPS + 1;
            flag = 1;
            lastTimer = timerMins;

            if (currentScreen == 1)
            {
              //SimbleeForMobile.updateValue(text2ID, STEPS);
            }
            if (currentScreen == 2)
            {
              SimbleeForMobile.updateValue(text6ID, STEPS);
            }
          }
           else if (totave[i] > threshhold && flag == 1)
          {
           //do nothing
          }
    }
     if (totave[i] < threshhold  && flag == 1)
     {
       flag=0;
       lastStepDebounceTime = millis();
     }
   }




//this code would be used if you had an extra analog pin for a temp sensor:
/*
    float tempVoltage;

  // First we'll measure the voltage at the temperature analog pin. Normally
  // we'd use analogRead(), which returns a number from 0 to 1023.
  // Here we've written a function (further down) called
  // getVoltage() that returns the true voltage (0 to 5 Volts)
  // present on an analog input pin.

//  tempVoltage = getVoltage(tempPin);

  // Now we'll convert the voltage to degrees Celsius.
  // This formula comes from the temperature sensor datasheet:
```

```
  float tempC = (tempVoltage - 0.5) * 100.0;

   // While we're at it, let's convert degrees Celsius to Fahrenheit.
  // This is the classic C to F conversion formula:

  float tempF = (tempC * (9.0/5.0)) + 32.0;
  CURRENTTEMP = tempF;

  if (currentScreen == 1)
  {
    if (stripSelection == 1){
            STRIPCHOICE = CURRENTTEMP;
            SimbleeForMobile.updateValue(text3ID, STRIPCHOICE);

    }
  }
  if (currentScreen == 2)
  {
            SimbleeForMobile.updateValue(text10ID, CURRENTTEMP);
  }
  */


//This is seeing if the button is switched and if it is looks for your pulse and calculates your
//heart rate:
        /*if (buttonRead == LOW)
        {
          findPulseStuff();
          LASTHEARTRATE = BPM;
        }*/

        findPulseStuff();
        LASTHEARTRATE = BPM;
//        SimbleeForMobile.updateValue(text2ID, LASTHEARTRATE);
        Serial.print("heart rate:");
        Serial.println(BPM); //for debugging
        delay(1);


//This code checks what screen you are on the app, and if it is the first screen it makes sure to
//put the right text into the middle based on what button you have pressed last:
  if (currentScreen == 1)
  {
    SimbleeForMobile.updateValue(text2ID, LASTHEARTRATE);
    if (stripSelection == 1){
            STRIPCHOICE = caloriesTotal;
            SimbleeForMobile.updateValue(text3ID, STRIPCHOICE);

    }
```

```
 }


    if (currentScreen == 1)
    {
       if (stripSelection == 2)
       {
         STRIPCHOICE = LASTHEARTRATE;
         SimbleeForMobile.updateValue(text3ID, STRIPCHOICE);
       }
           if (stripSelection != (0 || 1 || 2))
         {
          STRIPCHOICE = 0;
         }
    }


//This last function turns on the vibe motors if you haven't been active for more than 60 mins,
//and won't stop until you active again:
      if (LASTACTIVITY >= 60)  //if you have been inactive for an hour
        {
         digitalWrite(vibePin, HIGH); //set off the vibe motor
         digitalWrite(vibePin2, HIGH); //set off the vibe motor
        }
      else if (LASTACTIVITY < 60)
        {
         digitalWrite(vibePin, LOW);
         digitalWrite(vibePin2, LOW);
        }


      // This function must be called regularly to process UI events.
      SimbleeForMobile.process();
}



//Funtion to change the RGB LED to a specific color:
void RGB_Show(uint8_t r, uint8_t g, uint8_t b, uint32_t ledPin)
{
   uint8_t rgb[3] = {g, r, b};
   uint8_t *p = rgb;
   uint8_t *end = p + 3;

   while (p < end)
   {
      uint8_t pix = *p++;
      for (uint8_t mask = 0x80; mask; mask >>= 1)
      {
```

```
        if (pix & mask)
        {
          // T1H 760ns
          NRF_GPIO->OUTSET = (1UL << ledPin);
          NRF_GPIO->OUTSET = (1UL << ledPin);
          NRF_GPIO->OUTSET = (1UL << ledPin);

          // T1L 660ns
          NRF_GPIO->OUTCLR = (1UL << ledPin);
        }
        else
        {
          // T0H 380ns
          NRF_GPIO->OUTSET = (1UL << ledPin);

          // T0L 840ns
          NRF_GPIO->OUTCLR = (1UL << ledPin);
          NRF_GPIO->OUTCLR = (1UL << ledPin);
          NRF_GPIO->OUTCLR = (1UL << ledPin);
          NRF_GPIO->OUTCLR = (1UL << ledPin);
        }
      }
    }
    NRF_GPIO->OUTCLR = (1UL << ledPin);
}


//Function that takes analog values from a temp sensor and calculates the actual voltage output:
float getVoltage(int pin)
{
  // This function has one input parameter, the analog pin number
  // to read. You might notice that this function does not have
  // "void" in front of it; this is because it returns a floating-
  // point value, which is the true voltage on that pin (0 to 5V).

  // You can write your own functions that take in parameters
  // and return values. Here's how:

    // To take in parameters, put their type and name in the
    // parenthesis after the function name (see above). You can
    // have multiple parameters, separated with commas.

    // To return a value, put the type BEFORE the function name
    // (see "float", above), and use a return() statement in your code
    // to actually return the value (see below).

    // If you don't need to get any parameters, you can just put
    // "()" after the function name.
```

```
   // If you don't need to return a value, just write "void" before
   // the function name.

  // Here's the return statement for this function. We're doing
  // all the math we need to do within this statement:


  return ((analogRead(pin) * (3.3/1023)));

  // This equation converts the 0 to 1023 value that analogRead()
  // returns, into a 0.0 to 5 value that is the true voltage
  // being read at that pin.
}


//Function that gets your heart rate from the pulse sensor:
void findPulseStuff ()
{
  for (int pulseFlag = 0; pulseFlag < 10; pulseFlag++)
    {

          Signal = analogRead(pulsePin);              // read the Pulse Sensor
          //Serial.println(Signal);
          sampleCounter += 1;                         // keep track of the time in mS with this variable
          int N = sampleCounter - lastBeatTime;       // monitor the time since the last beat to avoid noise

       //  find the peak and trough of the pulse wave
          if(Signal < thresh && N > (IBI/5)*3){       // avoid dichrotic noise by waiting 3/5 of last IBI
             if (Signal < T){                         // T is the trough
                T = Signal;                           // keep track of lowest point in pulse wave
              }
             }

          if(Signal > thresh && Signal > P){          // thresh condition helps avoid noise
             P = Signal;                              // P is the peak
             }                                        // keep track of highest point in pulse wave

        //  NOW IT'S TIME TO LOOK FOR THE HEART BEAT
        // signal surges up in value every time there is a pulse
       if (N > 250){                                  // avoid high frequency noise
         if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)*3) ){
           Pulse = true;                              // set the Pulse flag when we think there is a pulse
           RGB_Show(0x00, 0x00, 0x00, LEDPIN);
           IBI = sampleCounter - lastBeatTime;        // measure time between beats in mS
           lastBeatTime = sampleCounter;              // keep track of time for next pulse

               if(firstBeat){                         // if it's the first time we found a beat, if firstBeat == TRUE
                  firstBeat = false;                  // clear firstBeat flag
                  return;                            // IBI value is unreliable so discard it
```

```
            }
          if(secondBeat){                     // if this is the second beat, if secondBeat == TRUE
            secondBeat = false;               // clear secondBeat flag
              for(int i=0; i<=9; i++){        // seed the running total to get a realisitic BPM at startup
                  rate[i] = IBI;
                  }
            }

        // keep a running total of the last 10 IBI values
        word runningTotal = 0;                // clear the runningTotal variable

        for(int i=0; i<=8; i++){              // shift data in the rate array
             rate[i] = rate[i+1];             // and drop the oldest IBI value
             runningTotal += rate[i];         // add up the 9 oldest IBI values
           }

        rate[9] = IBI;                        // add the latest IBI to the rate array
        runningTotal += rate[9];              // add the latest IBI to runningTotal
        runningTotal /= 10;                   // average the last 10 IBI values
        BPM = 60000/runningTotal;             // how many beats can fit into a minute? that's BPM!
        QS = true;                            // set Quantified Self flag
        // QS FLAG IS NOT CLEARED INSIDE THIS ISR
        }
      }

       if (Signal < thresh && Pulse == true){     // when the values are going down, the beat is over
          RGB_Show(redVal, greenVal, blueVal, LEDPIN);
          Pulse = false;                      // reset the Pulse flag so we can do it again
          amp = P - T;                        // get amplitude of the pulse wave
          thresh = amp/2 + T;                 // set thresh at 50% of the amplitude
          P = thresh;                         // reset these for next time
          T = thresh;
         }

       if (N > 2500){                         // if 2.5 seconds go by without a beat
          thresh = 512;                       // set thresh default
          P = 512;                            // set P default
          T = 512;                            // set T default
          lastBeatTime = sampleCounter;       // bring the lastBeatTime up to date
          firstBeat = true;                   // set these to avoid noise
          secondBeat = true;                  // when we get the heartbeat back
         }
       }

}



//Function you can run to find that average X and Y vallues from your accelorometer, to make
```

```
//calculating your steps easier and more accurate:
void calibrate()
{
     float sum=0;
     float sum1=0;

   for (int i=0;i<100;i++)
   {
   xval[i]=float(analogRead(accXPin));

   sum=xval[i]+sum;
   }
   xavg=sum/100.0;

   Serial.println(xavg);

   for (int j=0;j<100;j++)
   {
   yval[j]=float(analogRead(accYPin));

   }
   yavg=sum1/100.0;

   Serial.println(yavg);

}


//Function for the first screen of the phone app:
void screen1()
{

      uint16_t height = SimbleeForMobile.screenHeight;
      uint16_t width = SimbleeForMobile.screenWidth;
      // These variable names are long...let's shorten them. They allow us to make
      // an interface that scales and scoots appropriately regardless of the screen
      // orientation or resolution.

      color_t lightBlue = rgb(0,85,250);

      SimbleeForMobile.beginScreen(lightBlue, PORTRAIT);

      rectangle1ID = SimbleeForMobile.drawRect(
                  (width/2) - 100,      // x location
                  (height/4) - 10,      // y location
                  200,                  // width of rectangle
                  50,                  // height of rectangle
                  WHITE);              // color of rectangle */
```

```
rectangle2ID = SimbleeForMobile.drawRect(
            (width/2) - 100,      // x location
            (height/2),          // y location
            200,                 // width of rectangle
            50,                  // height of rectangle
            WHITE);              // color of rectangle

text1ID = SimbleeForMobile.drawText(
            (width/2) - 100 ,      // x location
            (height/4) - 75,       // y location
            "Heart rate:",             // shown text
            WHITE,                 // color of text
            36);                // size of text from 8 to 36

text2ID = SimbleeForMobile.drawText(
            (width/2) - 30 ,      // x location
            (height/4),       // y location
            BPM,             // shown text
            BLACK,                 // color of text
            30);                // size of text from 8 to 36

text3ID = SimbleeForMobile.drawText(
            (width/2) - 20 ,      // x location
            (height/2) + 10,       // y location
            STRIPCHOICE,             // shown text
            BLACK,                 // color of text
            30);                // size of text from 8 to 36

text3unitID = SimbleeForMobile.drawText(
            (width/2) + 30 ,      // x location
            (height/2) + 10,      // y location
            stripChoiceUnit,      // shown text
            BLACK,                // color of text
            30);                // size of text from 8 to 36


text4ID = SimbleeForMobile.drawText(
            (width/2) - 150,           // x location
            3*(height/4),       // y location
            "LEDs On/Off:",             // shown text
            WHITE,                 // color of text
            22);                // size of text from 8 to 36

button1ID = SimbleeForMobile.drawButton(
            (width/2) - 75,      // x location
            (height/2) - 75,      // y location
            150,                 // width of button
            "All Info",              // text shown on button
            BLACK,                // color of button
```

```
        BOX_TYPE);          // type of button

// Buttons, by default, produce only EVENT_PRESS type events. We want to also
// do something when the user releases the button, so we need to invoke the
// setEvents() function. Note that, even though EVENT_PRESS is default, we
// need to include it in setEvents() to avoid accidentally disabling it.
SimbleeForMobile.setEvents(button1ID, EVENT_PRESS | EVENT_RELEASE);

addCalBtnID = SimbleeForMobile.drawButton(
        3*(width/4) - 60,     // x location
        (height) - 75,     // y location
        120,               // width of button
        "Add Calories",            // text shown on button
        BLACK,             // color of button
        BOX_TYPE);          // type of button

SimbleeForMobile.setEvents(addCalBtnID, EVENT_PRESS | EVENT_RELEASE);

LEDswitchID = SimbleeForMobile.drawSwitch(
        (width/2) - 110,      // x location
        3*(height/4) + 50,     // y location
        GREEN);            // color of switch

// Syntax for specifying segment titles; odd names and values used to show
//  that segment text and title variable names are completely arbitrary.
const char* activity = "Last Activity";
const char* cals = "Calories";
const char* heart = "Heart Rate";

const char* const strip[3] = {activity, cals, heart};
segment1ID = SimbleeForMobile.drawSegment(
        0,                  // x location
        3*(height/4) - 60,         // y location
        width,                // width of segment
        strip,                // segment context
        3,                  // number of segments
        BLACK);                // color of segment

SimbleeForMobile.setEvents(segment1ID, 0 | 1 | 2);

int16_t CalorieAdd = 0;

InputTextFldID = SimbleeForMobile.drawTextField(
        3*(width/4) - 45,             // x location
        (height) - 120,             // y location
        90,               // text field width
        CalorieAdd,           // value entered
        "Enter Value",      // placeholder text
        BLACK,            // text color
```

```
                WHITE);   // background color

        SimbleeForMobile.setEvents(InputTextFldID, EVENT_PRESS | EVENT_RELEASE);

        SimbleeForMobile.endScreen();

        //These are updated outside of the screen in case anything changes while you are looking
        //at the screen:
        SimbleeForMobile.updateValue(text2ID, BPM);
        SimbleeForMobile.updateValue(text3ID, STRIPCHOICE);
        SimbleeForMobile.updateText(text4ID, "LEDs On/Off:");
}


//Function for the second screen of the phone app:
void screen2()
{
        uint16_t height = SimbleeForMobile.screenHeight;
        uint16_t width = SimbleeForMobile.screenWidth;
        color_t lightBlue = rgb(0,85,250);

        Serial.print("SCRN2 cal: ");
        Serial.println(caloriesTotal);

        SimbleeForMobile.beginScreen(lightBlue, PORTRAIT);

        text5ID = SimbleeForMobile.drawText(
                10,     // x location
                (height/14),      // y location
                "Steps Today:",           // shown text
                BLACK,              // color of text
                30);              // size of text from 8 to 36

        text6ID = SimbleeForMobile.drawText(
                20,     // x location
                2*(height/14),      // y location
                STEPS,          // shown text
                WHITE,              // color of text
                26);              // size of text from 8 to 36

        text7ID = SimbleeForMobile.drawText(
                10,          // x location
                3*(height/14),            // y location
                "Last Activity:",           // shown text
                BLACK,              // color of text
                30);              // size of text from 8 to 36

        lastActTextID = SimbleeForMobile.drawText(
                20,          // x location
```

```
                4*(height/14),      // y location
                LASTACTIVITY,           // shown text
                WHITE,              // color of text
                26);            // size of text from 8 to 36

        activeUnitID = SimbleeForMobile.drawText(
                60,         // x location
                4*(height/14),      // y location
                activityUnit,           // shown text
                WHITE,              // color of text
                26);            // size of text from 8 to 36



        //If you wanted to show temp sensor info (if you had it):
/*
        text9ID = SimbleeForMobile.drawText(
                10,         // x location
                9*(height/14),      // y location
                "Current Temperature:",         // shown text
                BLACK,              // color of text
                30);            // size of text from 8 to 36

        text10ID = SimbleeForMobile.drawText(
                20,         // x location
                10*(height/14),     // y location
                CURRENTTEMP,            // shown text
                WHITE,              // color of text
                26);            // size of text from 8 to 36'



        text10UnitID = SimbleeForMobile.drawText(
                50,         // x location
                10*(height/14),     // y location
                tempUnit,           // shown text
                WHITE,              // color of text
                26);            // size of text from 8 to 36
  */

        text11ID = SimbleeForMobile.drawText(
                10,         // x location
                5*(height/14),      // y location
                "Last Heart Rate",          // shown text
                BLACK,              // color of text
                30);            // size of text from 8 to 36

        text12ID = SimbleeForMobile.drawText(
                20,         // x location
```

```
            6*(height/14),      // y location
            LASTHEARTRATE,          // shown text
            WHITE,              // color of text
            26);            // size of text from 8 to 36

    text12UnitID = SimbleeForMobile.drawText(
            75,         // x location
            6*(height/14),      // y location
            heartUnit,          // shown text
            WHITE,              // color of text
            26);            // size of text from 8 to 36


    text13ID = SimbleeForMobile.drawText(
            10,         // x location
            7*(height/14),      // y location
            "Total Calories Today:",        // shown text
            BLACK,              // color of text
            30);            // size of text from 8 to 36

    calTextFldID = SimbleeForMobile.drawText(
            20,         // x location
            8*(height/14),  // y location
            caloriesTotal,  // shown text
            WHITE,          // color of text
            26);


    button3ID = SimbleeForMobile.drawButton(
            (width/2) - 75,             // x location
            13*(height/14),     // y location
            150,            // width of button
            "Back",             // text shown on button
            BLACK,              // color of button
            BOX_TYPE);          // type of button

    SimbleeForMobile.setEvents(button3ID, EVENT_PRESS | EVENT_RELEASE);

    SimbleeForMobile.endScreen();

    //These are updated outside of the screen in case anything changes while you are looking
    //at the screen:
    SimbleeForMobile.updateValue(lastActTextID, LASTACTIVITY);
    SimbleeForMobile.updateText(activeUnitID, "mins");
    SimbleeForMobile.updateValue(calTextFldID, caloriesTotal);
    SimbleeForMobile.updateValue(text6ID, STEPS);
//      SimbleeForMobile.updateValue(text10ID, CURRENTTEMP);

}
```

```
// ui() is a SimbleeForMobile specific function which handles the specification
// of the GUI on the mobile device the Simblee connects to.
void ui()
{
	// color_t is a special type which contains red, green, blue, and alpha
	// (transparency) information packed into a 32-bit value. The functions rgb()
	// and rgba() can be used to create a packed value.
	color_t lightBlue = rgb(0,85,250);
	Serial.println("UICALLED");


	//The ui() fuction is called each time you want to change screens, and these are
	//the statements that see which screen to be on and to draw that screen:
	if (currentScreen == SimbleeForMobile.screen) return;

	currentScreen = SimbleeForMobile.screen;

	if (currentScreen ==  1)
		screen1();
	else
		screen2();

}



//This is another function you need when making a SimbleeForMobile project, it is the
//function that gets called each time an event from the app happens (such as a button push):
void ui_event(event_t &event)
{
	printEvent(event); //for debugging

	//Each statement specifies what to do when something specific happens from the app:

	if (event.id == InputTextFldID)
	{
		Add = event.value;
	}

	else if (event.id == button1ID)
	{
		if (event.type == EVENT_RELEASE)
		{
			SimbleeForMobile.showScreen(2);
		}
	}
```

```
else if (event.id == addCalBtnID)
{
	if (event.type == EVENT_PRESS)
	{
		caloriesTotal = caloriesTotal + Add;
		Serial.print("EVENT cal: ");
		Serial.println(caloriesTotal);
	}
	if (event.type == EVENT_RELEASE)
	{
		SimbleeForMobile.updateValue(InputTextFldID, 0);

	}
}

else if (event.id == button3ID)
{
	if (event.type == EVENT_RELEASE)
	{
		SimbleeForMobile.showScreen(1);
		stripSelection = 4;
	}
}

else if (event.id == LEDswitchID)
{
 if (event.value == 1)
 {
  leds = 0;
 }
 if (event.value == 0)
 {
  leds = 1;
 }

}

else if (event.id == segment1ID)
{
 stripSelection = event.value;
	uint16_t width = SimbleeForMobile.screenWidth;
	if (event.value == 0)
	{
		STRIPCHOICE = LASTACTIVITY;
		stripChoiceUnit = activityUnit;
		SimbleeForMobile.updateValue(text3ID, STRIPCHOICE);
		SimbleeForMobile.updateX(text3ID, (width/2) - 10);
		SimbleeForMobile.updateText(text3unitID, stripChoiceUnit);
	}
```

```
            if (event.value == 1)
            {
                STRIPCHOICE = caloriesTotal;
                stripChoiceUnit = " ";
                SimbleeForMobile.updateValue(text3ID, STRIPCHOICE);
                SimbleeForMobile.updateX(text3ID, (width/2) - 20);
                SimbleeForMobile.updateText(text3unitID, stripChoiceUnit);
            }
            if (event.value == 2)
            {
                STRIPCHOICE = LASTHEARTRATE;
                stripChoiceUnit = heartUnit;
                SimbleeForMobile.updateValue(text3ID, STRIPCHOICE);
                SimbleeForMobile.updateX(text3ID, (width/2) - 20);
                SimbleeForMobile.updateText(text3unitID, stripChoiceUnit);
            }
        }

}



//Fuction that runs when you connect your phone to the projected app:
void SimbleeForMobile_onConnect()
{
    currentScreen = -1;
    RGB_Show(0x00,0xff,0x00, LEDPIN);
    delay(100);
}


//Function for debugging purposes:
void printEvent(event_t &event)
{
    Serial.println("Event:");
    Serial.print("  ID: ");
    Serial.println(event.id);

    Serial.print("   Type: ");
    Serial.println(event.type);

    Serial.print("   Value: ");
    Serial.println(event.value);

    Serial.print("   Text: ");
    Serial.println(event.text);

    Serial.print("   Coords: ");
    Serial.print(event.x);
```

```
        Serial.print(",");
        Serial.println(event.y);

        Serial.println(STEPS);

//      Serial.print("Temp Read: ");
//      Serial.println(analogRead(tempPin));

//      Serial.print("Temp Display: ");
//      Serial.println(CURRENTTEMP);
}
```