

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Karl Meldorf**

**Terviseandmete jälgimine MySignals  
arenduskomplektiga**

**Bakalaureusetöö (9 EAP)**

Juhendaja: Alo Peets

Tartu 2020

## **Terviseandmete jälgimine MySignals arenduskomplektiga**

### **Lühikokkuvõte:**

Käesoleva bakalaureusetöö eesmärk on luua rakendus, mis kasutades MySignals'i arendusplaati ja sensoreid võimaldab mõõta kasutaja vere hapniku küllastatuse taset, vererõhku ning kehatemperatuuri. Mõõdetud tulemused salvestatakse loodavas veebirakenduses, kus kasutaja saab näha enda mõõtmistulemusi. Töö sekundaarne eesmärk on näidata, kuidas kasutada MySignals'i platvormi. Töö esimeses peatükis antakse ülevaade asjade interneti olulisusest ning tuuakse kaks näidet, kuidas oleks asjade internetist kasu tervishoius. Teises peatükis tutvustatakse töös kasutatud riistvara komponente. Kolmandas peatükis kirjeldatakse loodud rakendust ja selle osasid. Töö tulemusena valmis kaks Arduino juhtprogrammi ja veebirakendus, mis täitsid töös seatud eesmärgid.

**Võtmesõnad:** Arduino, MySignals, asjade internet, terviseandmed, Bluetooth

**CERCS:** P170 (Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine)

## **Health data monitoring with MySignals development kit**

### **Abstract:**

The aim of this bachelor thesis is to create an application, which can measure the user's blood oxygen saturation, blood pressure and body temperature using MySignals development board and sensors. Measured results are saved in the created web application, where the user can also see his/her measurements. The second aim of the thesis is to give an example how to use the MySignals platform. The first chapter gives an overview of Internet of Things and describes two examples in which Internet of Medical Things can be beneficial. In the second chapter used hardware components are described. Third chapter describes the created application and its parts. As a result of the thesis two Arduino programs and a web application were created, which achieved the proposed aim.

**Keywords:** Arduino, MySignals, Internet of Things, health data, Bluetooth

**CERCS:** P170 (Computer science, numerical analysis, systems, control)

## Sisukord

Sissejuhatus .....	4
1. Asjade internet ja tervishoid .....	5
1.1 Asjade internet .....	5
1.2 Asjade internet tervishoius .....	5
1.3 Kõrgvererõhutõbi ja vererõhu monitooring .....	6
1.4 Varajane hoiatusskoori süsteem .....	7
2. Riistvara .....	9
2.1 Arduino Uno .....	9
2.2 MySignals'i platvorm .....	10
2.3 WiFi moodul ESP8266 .....	11
2.4 Bluetooth moodul BLE112 .....	12
2.5 Vererõhu mõõtmisensor .....	12
2.6 Pulssoksümeeter ehk SpO2 .....	13
2.7 Temperatuurisensor DS18B20 .....	14
3. Tarkvara .....	15
3.1 MySignals teegid .....	15
3.2 Suhtlus WiFi mooduliga .....	15
3.3 Bluetooth suhtlus .....	17
3.4 Kehatemperatuuri mõõtmine .....	18
3.5 Loodud Arduino juhtprogramm .....	20
3.6 Arendusprotsess .....	22
3.7 Loodud veebirakendus .....	23
4. Kokkuvõte .....	25
Viidatud kirjandus .....	26
Lisad .....	29
I. Rakenduse lähtekoodide repositooriumid .....	29
II. Terminid .....	30
III. Litsents .....	32

## Sissejuhatus

Tehnoloogia areng - elektroonikaseadmete väiksemaks ja töökindlamaks muutumine on toonud suure populaarsuse kasvu asjade internetile (inglise keeles *Internet of Things*). Ka meditsiinivaldkond vaatab järjest enam IoT rakendamise võimaluste poole, sest arenenud riikides on probleemiks üha vananev rahvastik ning sellest tulenevalt suurenenud kulutused tervishoiule. Probleemi võib leevendada laialdasem tervisenäitajate jälgimine kodustes tingimustes ning selleks vajalike rakenduste ja sensorite arendamine.

Sensorite abil tehtud mõõtmistulemusi on võimalik saata otse meditsiinitöötajale või neid talletada ja analüüsida. Nii saab kasutaja pidevat tagasisidet oma tervise seisundi kohta ja on võimalik varakult märgata patsiendi tervise seisundi halvenemist. Kehal kantavad seadmed, mis mõõdavad näiteks kasutaja pulssi, tehtud samme ja unekvaliteeti, on kogunud viimastel aastatel palju populaarsust.

Antud töö eesmärk on luua süsteem, mis kasutades Arduino mikrokontrollerit ja MySignals'i arenduskomplekti võimaldab kasutajal mõõta tervisenäitajaid ja talletada neid loodavas veebirakenduses. Veebirakendus võimaldab lisaks andmete talletamisele kasutada ka varajase hoiatusskoori (*Early Warning Score* ehk EWS) süsteemi, mille abil on võimalik hinnata patsiendi tervise seisundi tõsidust. Kuigi praegu kasutatakse EWS süsteemi peamiselt haiglates, siis Silcock ja teised [1] on näidanud, et selle kasutamisest võib kasu olla ka kodustes tingimustes. Lisaks on antud töö ka illustreeriv näide, kuidas kasutada MySignals'i platvormi.

Töö on jaotatud kolmeks peatükiks. Esimeses peatükis räägitakse IoT võimalustest tervishoius. Teises peatükis saab ülevaate töös kasutatud riistvara komponentidest. Kolmandas peatükis tuuakse kirjutatud tarkvarast koodinäiteid ning kirjeldatakse, kuidas loodud rakendus toimib.

## 1. Asjade internet ja tervishoid

Antud peatükis antakse ülevaade, mis on *Internet of Things* ehk asjade internet ja mis on *Internet of Medical Things* ehk meditsiiniliste asjade internet. Lisaks tuuakse kaks näidet sellest, kuidas patsientidel IoT lahendusest kasu oleks.

### 1.1 Asjade internet

Asjade internet - termin võeti kasutusele 1990-ndatel aastatel ning selle kohta ei ole ühte kindlat aktsepteeritud definitsiooni, kuid üks võimalik määratlus oleks järgmine:

„Avatud ja laiaulatuslik võrgustik intelligentsetest seadmetest, millel on võimekus enda tööd automaatselt organiseerida ning jagada informatsioon, andmeid ja ressursse, reageerides keskkonna muutustele.“ [2]

Teiste sõnadega tähendab see, et asjade internet tähistab seadmeid, mis kas koguvad andmeid füüsilise keskkonna kohta ja/või on võimelised seda ka mõjutama. Interneti teel on inimesel võimalik kontrollida nende seadmete olekut ja/või seadmete kogutud andmeid. International Data Corporation (IDC) tehtud uuring ennustab, et IoT seadmete hulk kasvab stabiilselt ning aastaks 2025 on maailmas 41,6 miljardit seadet, mis kokku genereerivad aasta jooksul 79,4 Zettabaiti andmeid [3].

Esimene Internetti ühendatud seade oli Coca-Cola joogiautomaat Carnegie Melon Ülikoolis 1980-ndate alguses [2]. Programmeerijad, kes töötasid automaadist mitu korrust kõrgemal, kirjutasid programmi, millega said üle interneti kontrollida joogiautomaadi staatust ja otsustada, kas seal on külm jook olemas ehk kas neil tasub jalutada mitu korrust alla automaadini. Sellest ajast on tehnoloogia palju arenenud ning kaks suurimat IoT rakendamise valdkonda, millele ennustatakse kõige suuremat kasvu on tootmine ja tervishoid [4].

### 1.2 Asjade internet tervishoius

Tervishoid on asjade interneti rakendamiseks üks atraktiivsemaid valdkondi. IoT võimaldab luua palju erinevaid meditsiini rakendusi nagu tervise kaugjälgimine, krooniliste haiguste jälgimine ja eakate hooldamine. Samuti on üheks võimalikuks rakenduseks kodusel ravil

olevate patsientide jälgimine, et kontrollida, kas määratud ravi toimib nagu plaanitud [4]. Al-Fuqaha ja teised on ennustanud, et meditsiiniliste asjade interneti mõju maailma majandusele aastaks 2025 on 1,1–2,5 triljonit dollarit. Kogu IoT valdkonnale ennustatakse, et aastaks 2025 tõstab see maailma majanduse kogutoodangut vahemikus 2,7 kuni 6,2 triljonit dollarit [5]. Seega moodustaks tervishoid 40% kogu IoT tehnoloogiast saadavast majanduslikust kasust.

### 1.3 Kõrgvererõhutõbi ja vererõhu monitooring

Maailma Terviseorganisatsiooni andmetel oli 2008. aastal kõrgvererõhutõbi maailmas 40% inimestest, kes olid vanemad kui 25. Südame veresoonehaigused põhjustavad umbes pooled kõigist surmaga lõppevatest haigusjuhtudest [6]. Peamine uuring, mida kõrgvererõhutõve puhul läbi viiakse on vererõhu 24-tunni monitooring. Tartu Ülikooli kliinikum kirjutab vererõhu 24-tunni monitooringu kohta oma kodulehel [7] järgmist:

*„Vererõhu 24-tunni monitooring võimaldab hinnata patsiendi vererõhu taset väljaspool arstikabinetti, patsiendile tavapärasel keskkonnas ja tegevuste juures. Hästi on teada vererõhu kõikumine ärevuse, hirmu ja pingel korral – seetõttu on võimalik, et arsti kabinetis mõõdetud kõrge vererõhk ei pruugi samasugune olla kodustes tingimustes. Uuring on vajalik hüpertooniatõve diagnoosi välistamiseks või kinnitamiseks kõrge vererõhu näitade korral ning juba varem diagnoositud hüpertooniatõve ravitulemuste hindamiseks. Vererõhu näidud salvestatakse ühe ööpäeva jooksul. Aparaat mõõdab vererõhku päeval iga 15 minuti järel ja öösel iga 30 minuti järel.“*

Praegu toimub see protsess enamasti nii, et vererõhumõõtja paneb patsiendi käele arst, tulemused salvestatakse aparati. Kui patsient on tagasi arsti juures võetakse aparadist näidud. Seejärel tehakse näitade põhjal automaatne analüüs ja arst annab oma hinnangu. Seda protsessi oleks võimalik automatiseerida ning patsiendile saaks analüüsi pakkuda ilma, et patsient lahkuks kodust. Daniel Ruiz-Fernández ja teised pakuvad oma uurimistöös [8], et krooniliste vererõhuhaigete raviprotsessi võiks muuta rohkem IoT lahenduste keskseks, toetudes lisaks pidevale vererõhu mõõtmisele ka patsiendi kehakaalule, aktiivsusele ja keskkonna temperatuurile. Neid andmeid koos analüüsides saaksid arstid pakkuda paremat ravi - korrigeerida ravimite annuseid ning jälgida pikema aja jooksul toimunud muutusi. Lisaks annaks see parema ülevaate ka patsientidele enda tervises seisundist ning vähem arstivisiite tähendaks kokku hoitud aega ja raha.

## 1.4 Varajane hoiatusskoori süsteem

Üks terviseandmete mõõtmise ja jälgimise peamine funktsioon on varajane tervise halvenemise tuvastamine. Varajase hoiatusskoori (*Early Warning Score* ehk EWS) süsteemi eesmärk on tuvastada ja ennustada patsiendi tervises seisundi halvenemist haiglas. Süsteem pakuti välja Morgani ja teiste poolt aastal 1997 [9] ning selleks monitooritakse pidevalt patsiendi elulisi näitajaid. Tihti tehakse arvutused paberi ja pliiatsiga. Kuna tervishoiu ressursid on piiratud ja haiglakoha hind kõrge, siis ei pruugi patsient olla haiglas terve ravi jooksul ning see on loonud vajaduse varajase hoiatusskoori süsteemi kasutamiseks ka kodus [10].

Varajase hoiatusskoori arvutamiseks mõõdetakse patsiendi elulisi näitajaid nagu pulss, hingamissagedus, kehatemperatuur, vererõhk ja vere hapnikuküllastatuse tase. Igale näitajale määratakse skoor vastavalt sellele, kui palju see erineb määratud normaalsest vahemikust ning kõikide tulemuste summa näitab patsiendi riskitaset. Tabel 1 näitab varajase hoiatusskoori hindamisjuhendit.

Tabel 1. Varajase hoiatusskoori arvutamise mudel. [11]

Füsioloogiline parameeter	3	2	1	0	1	2	3
Hingamisagedus (hingamiste arv minutis)		0-8		9-14	15-20	21-29	30+
SpO2 (%)		0%-84%	85%-89%	95%-100%			
Kehatemperatuur		0-35		35.1-38.0		38.1-39.5	39.6+
Süstoolne vererõhk (mmHg)	0-69	70-80	81-100	101-149	150-169	170-179	180+
Pulss (lööke minutis)	0-39	40-50	51-59	60-100	101-110	111-129	130+
Teadvusetase*				A	V	P	U

(\*A=Alert, V=response to voice, P=response to pain, U=unresponsive)

Näitajate kokku liitmisel saadav summa jääb vahemikku 0-18. Patsiendid on madala haigusriskiga, kui summa on 4 või vähem, keskmise riskiga, kui summa on 5-6 või üksiku

näitaja skoor on 3. Patsient, kelle EWS skoor on 7 või rohkem on kõrge riski kategoorias ja vajab pidevat monitoorimist ning võimalik, et ka intensiivravi [11].

Seega peab kodustes tingimustes süsteem hoiatuse andma juba palju varem ja kindlasti mitte alles siis, kui skoor on 7 või rohkem. Samuti tuleb arvestada, et patsientide normaalsed vahemikud võivad olla erinevad ning EWS ei asenda professionaalset hinnangut. Vaatamata sellele järeldasid Silcock ja teised, et kõrgem EWS skoor kodus viibivatel patsientidel on piisavalt hea indikaator ja selle arvutamisest võib olla kasu patsientide tervise halvenemise tuvastamisel. Varajase hoiatusskoori süsteem võimaldaks patsientidel saada varakult vajalikku meditsiinilist abi [1]. Kirjeldatud süsteem on implementeeritud loodud veebirakenduses, kuhu automaatselt jõuavad vererõhk, kehatemperatuur, pulss. Manuaalselt on vajalik patsiendi hingamissageduse ja teadvusetaseme sisestamine.

## 2. Riistvara

Käesolevas peatükis annab autor ülevaate töös kasutatud riistvara komponentidest. Komponentideks on Arduino Uno, kolm töös kasutatud sensorit ja MySignals'i arendusplaat koos selle küljes olevate WiFi ja Bluetooth moodulitega.

### 2.1 Arduino Uno

Arduino on vabavaraline platvorm, mida kasutatakse elektroonikaprojektides. Arduino koosneb nii füüsilisest programmeeritavast trükiplaadist koos mikrokontrolleriga kui ka integreeritud arenduskeskkonnast (*Integrated Development Environment* ehk IDE), mille abil saab kirjutada Arduinole programmikoodi, selle kompileerida ning mikrokontrollerile laadida [12].

Arduino Uno (Joonis 1) on arendusplaat, mis põhineb ATmega328P mikrokontrolleril. “Uno” tähendab itaalia keeles number ühte ning see nimi valiti tähistamiseks Arduino IDE 1.0 väljalaskmist. Arduino Uno oli ka esimene USB ühendusega Arduino arendusplaat. Plaadil on 14 digitaalset sisend-/väljundpesa ja 6 analoogsisendit [13]. ATmega328 kiibil on kolme sorti mälu: 32 kB *Flash* mälu ehk püsimälu, millele laetakse programmikood, 2 KB SRAM muutmälu, kus hoitakse väärtusi programmi jooksmise ajal ning 1 KB EEPROM püsimälu, kus hoitud muutujad jäävad alles ka peale Arduino taaskäivitamist [14].



Joonis 1. Arduino Uno [13].

## 2.2 MySignals'i platvorm

MySignals on arendusplatvorm terviseandmete kogumiseks mõeldud sensorite ja rakenduste arendamiseks. MySignals jaotab oma platvormi kaheks: tarkvara platvorm ja riistvara platvorm. Tarkvara platvorm, mis tähendab juba töövalmis seadme (vt joonis 2) kasutamist, mille külge on võimalik ühendada nende kõiki olemasolevaid sensoreid. See tähendab, et tervisenäitajate mõõtmisega on võimalik alustada koheselt. Mõõdetud tulemused salvestatakse seadmes ja pilves. Sealjuures on vaja andmete API abil kätte saamiseks osta ettevõttelt arendaja litsents, mille hind jääb vahemikku 199 kuni 599 eurot aastas (<http://www.libelium.com/cloud-services/mysignals-cloud/>). Tarkvara platvormi antud töös ei kasutata. Teine on riistvara platvorm, mis sisaldab arendus plaati, millel on 11 ühenduskohta erinevate sensorite jaoks, Bluetooth moodul, WiFi moodul ja TFT ekraan. Joonisel 2 on näha tarkvara platvormi seadet ja riistvara platvormi arendusplaati. MySignals'i arendusplaat on vaja ühendada Arduino Unoga.



Joonis 2. MySignals'i tarkvara platvormi seade ja riistvara platvormi arendusplaat [15].

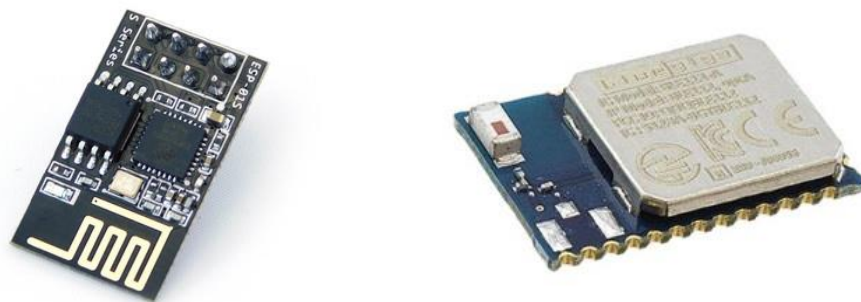
MySignals pakub oma kodulehel ja dokumentatsioonis arendajatele ka Androidi ja iOS API't mobiilirakenduse arendamiseks. Mobiilirakendus oleks võimaldanud paremat kasutajakogemust, sest nutitelefon on enamasti inimesega kaasas, seega oleks pidevam monitoorimine. Sellepärast oli töö algne eesmärk arendada mobiilirakendus Androidile. Sellega tegi autor ka algust, kasutades ettevõtte dokumentatsioonist alla laetavat API't ja koodinäiteid. Palju töötunde kulus põhjuste otsimiseks, miks antud koodinäitega ei olnud võimalik sensoritega ühendust luua. Põhjus leiti hiljem ettevõtte foorumist [16], kus kasutaja mobiilirakenduse näitekoodi mittetoimimise murede peale oli vastatud, et API arendamise ja

toetamisega ettevõtte enam ei tegele. Seega pakub MySignals oma kodulehel ja dokumentatsioonis vananenud infot.

Põhjus, miks arendajatele mobiilirakenduste arendamiseks API't ei pakuta on ilmselt äriiline, sest tarkvara ja teenuste müümine on võrreldes riistvara müümisega üldjuhul palju kasumlikum ehk kõrgema kasumimarginaaliga äri. Nüüd peavad arendajad kasutama ettevõtte enda mobiilirakendust, mis tähendab, et arendajatelt saab raha küsida pilveteenuse andmetele ligipääsuks.

### 2.3 WiFi moodul ESP8266

ESP8266 on mikrokiip, mis koosneb mikrokontrollerist ja WiFi vastuvõtjast. Kiipi võib kasutada nii välise WiFi moodulina, kasutades AT käsklusi üle mikrokontrolleri UART pesade, kui ka eraldi seisva WiFi võimekusega mikrokontrollerina [17].



Joonis 3. Vasakul ESP8266 WiFi moodul [18] ning paremal BLE112 Bluetooth moodul [19].

AT käskude kogum (*AT Command set*) on standard modemite kontrollimiseks, arendatud Dennis Hayes'i poolt 1981. aastal. Tuntakse ka Hayes'i käskude kogumiku (*Hayes command set*) nime all [20]. Loomulikult saab vältida AT käskude tundmaõppimist ja kasutada hoopis vastavat teeki. Kuna teegid kulutavad palju mälu on töös kasutatud AT käsklusi otse. Tabelis 2 on välja toodud töös kasutatud käsud ja nende kirjeldused.

Tabel 2. AT käsud ja nende kirjeldused [21].

Käsud	Kirjeldus
AT	Testib kas moodul töötab
AT+CWMODE	WiFi režiim (1-station, 2-access point, 3-mõlemad)
AT+CWJAP	Ühine pääsupunktiga (AP-ga)
AT+CIPSTART	TCP või UDP ühenduse loomine
AT+CIPSEND	Andmete saatmine (parameeter andmete tähtede arv)
AT+CIPCLOSE	TCP või UDP ühenduse sulgemine

## 2.4 Bluetooth moodul BLE112

Sensoritega suhtluseks on MySignals'i plaadil olemas Bluetooth moodul BLE112, mis on mõeldud sensoritele ja seadmetele, millel on vajalik madal energiatarbimine. Madalaima energiatarbimisega unerežiimis tarbib moodul ainult 500 nanoampri jagu voolu ja on suuteline uuesti tööle hakkama vaid mõne saja mikrosekundiga [22]. MySignals pakub kuut erinevat Bluetooth sensorit: kaal, SpO<sub>2</sub>, vererõhumõõtja, glükomeeter, kehatemperatuuri mõõtja ja alarm-/hädaabinupp. Töös kasutatakse pulsi ja verehapniku ehk SpO<sub>2</sub> sensorit ning vererõhu mõõtmisseadet.

## 2.5 Vererõhu mõõtmissensor

Vererõhk on rõhk, mis tekib arteris südame töö tulemusena. Vererõhu mõõtmisel mõõdetakse kaks numbrit. Esiteks süstoolne rõhk ehk rõhk, kui süda kontraheerub ning teiseks diastoolne rõhk, mida mõõdetakse südamelöökide vahepeal [15].

Vererõhu mõõtmine toimub Bergeni [23] kirjelduse kohaselt järgmisel põhimõttel. Ümber käe ülaosa pannakse mansett, kuhu pumbatakse õhku kuni arteris ei pääse veri enam manseti alt läbi. Seejärel hakatakse mansetist õhku välja laskma kuni veri hakkab arteris jälle voolama ning arteri seinal tekib vibratsioon, mille sensor tuvastab. See tähendab, et rõhk on langenud alla inimese süstoolse vererõhu ning sellel hetkel toimub mõõtmine. Kui manseti rõhk väheneb veelgi ja arteris voolab veri jälle normaalselt, siis vibratsioon kaob. Sellel hetkel mõõdetakse diastoolne rõhk.

Vererõhk ei ole inimestel püsivalt sama, vaid muutub vastavalt keha vajadustele. Seda mõjutavad näiteks kehaasend, hingamine ning ka emotsionaalne seisund. Sellepärast on kõige parem mõõta vererõhku rahulikus olekus istudes või lamades [15]. Tabelis 3 on välja toodud vererõhuväärtuste klassifikatsioon. MySignals ütleb vererõhusensori vererõhu mõõtetäpsuseks  $\pm 3$  mmHg ja pulsi mõõtetäpsuseks  $< 5\%$ .

Tabel 3. Vererõhuväärtuste klassifikatsioon [24].

Nimi	Süstoolne (mmHg)	Diastoolne
<b>Optimaalne</b>	<b>&lt; 120</b>	<b>&lt; 80</b>
<b>Normaalne</b>	<b>120-129</b>	<b>80-84</b>
<b>Kõrge normaalne</b>	<b>130-139</b>	<b>85-89</b>
<b>I astme hüpertensioon ehk kõrgvererõhktõbi</b>	<b>140-159</b>	<b>90-99</b>
<b>II astme hüpertensioon</b>	<b>160-179</b>	<b>100-109</b>
<b>III astme hüpertensioon</b>	<b><math>\geq 180</math></b>	<b><math>\geq 110</math></b>
<b>Isoleeritud süstoolne hüpertensioon</b>	<b><math>\geq 140</math></b>	<b>&lt; 90</b>

## 2.6 Pulssoksümeeter ehk SpO2

SpO2 sensor võimaldab mitteinvasiivsel teel mõõta inimese pulssi ning arteriaalse hapniku küllastust funktsionaalses hemoglobiinis. Hapnikuga küllastatuse mõõtmistulemus saadakse hemoglobiini ja dioksühemoglobiini hulga veres määramise tulemusel. Mõõtmiseks kasutatakse kahte erineva lainepikkusega (660 nm ja 940 nm) valgust. Hemoglobiinis neeldub rohkem 660 nm lainepikkusega valgust ning dioksühemoglobiinis rohkem 940 nm lainepikkusega valgust. Lõpuks mõõdab fotodetektor mitteneeldunud valguse osa ja arvutab vere hapnikuküllastatuse taseme. Normaalseks hapnikuküllastatuse vahemikuks on 95-99%. Inimesel, kellel on hapniku omastamisega probleeme, on eeldatav vahemik 88-94% [15]. Pulssoksümeetri mõõtetäpsuse kohta ei ole MySignals'i dokumentatsioonis infot.



Joonis 4. Vasakul SpO2 sensor ja paremal vererõhu mõõtmisensor [15].

## 2.7 Temperatuurisensor DS18B20

Kehatemperatuuri mõõtmiseks on ka MySignals'il olemas vastav sensor, kuid edukalt saab kasutada ka populaarset DS18B20 üldotstarbelist temperatuurisensorit. Seda toodab Dallas Semiconductor Corp ja sensor on saadaval kahel erineval kujul: väikene takisti moodi sensor ja kaabliga versioon [25]. Kehatemperatuuri mõõtmiseks on sobilik kaabliga versioon.

DS18B20 temperatuurisensor on võimeline mõõtma temperatuure vahemikus  $-55^{\circ}\text{C}$  kuni  $125^{\circ}\text{C}$  mõõtetäpsusega  $\pm 0.5^{\circ}\text{C}$ . Sensori tulemuse esitamistäpsust on kasutajal võimalik seadistada 9. kuni 12. bitiseks. Vaikimisi on see seadistatud 12. bitiseks, mis annab arvutuslikuks täpsuseks  $0.0625^{\circ}\text{C}$  [25].



Joonis 5. Kahte tüüpi DS18B20 temperatuurisensendid [25].

### 3. Tarkvara

Selles peatükis annab autor ülevaate töös loodud ja kasutatud tarkvarast. Esimeses pooles vaadatakse MySignals'i teegi kasutamist ja koodinäidete abil näidatakse, kuidas kasutada WiFi ja Bluetooth moodulit. Teises pooles kirjeldatakse loodud süsteemi toimimist.

#### 3.1 MySignals teegid

Arendajatele Bluetoothi ja WiFi mooduli lihtsamaks kontrollimiseks pakub MySignals enda spetsiaalselt arendatud teeke. Teegid ei ole lisatud Arduino teekide haldajasse (*Arduino library manager*), vaid tuleb alla laadida MySignals'i platvormi dokumentatsioonist [15], kus vastav link "*Download the libraries*" paikneb viienda peatüki alguses. Teegi kasutusele võtmiseks tuleb see kõigepealt lahti pakkida ja seejärel Arduino arenduskeskkonnas valida Sketch -> Include library -> Add .ZIP library. Avanenud dialoogikastis tuleb valida ja lisada lahti pakitud kausta sisu. Peale seda saab lisatud teeke lisada nagu teisi teeke.

#### 3.2 Suhtlus WiFi mooduliga

MySignals'i arendusplaadi küljes oleva ESP8266 WiFi mooduliga suhtlus toimub üle Arduino UART (*Universal asynchronous receiver-transmitter*) pesade ning kogu suhtlus toimub kasutades AT käsklusi. See tähendab, et saadetavaid käsklusi ja mooduli tagasisidet on võimalik näha Arduino Serial monitoris. AT käskluste saatmiseks ja vastuste lugemiseks on töös kasutatud järgmist funktsiooni:

```
bool Controller::sendData(const char* command, const int timeout, boolean debug)
{
    String response = "";
    bool answer = false;

    while ( Serial.available() > 0) Serial.read();

    Serial.println(command); // käskluse saatmine
    long int time = millis();

    while( (time+timeout) > millis())
```

```

{
  while(Serial.available()) // vastuse lugemine
  {
    char c = Serial.read();
    response+=c;
  }
}

if(debug)
{
  MySignals.println(response.c_str());
}

return strstr(response.c_str(), "OK");
}

```

Funktsioon sendData saadab käskluse ja ootab parameetriga „timeout“ etteantud aja vastust. Lõpuks kontrollitakse, kas vastuses sisaldub sõne „OK“, mis tähendab, et käskluse täitmine oli edukas. Kasutades seda funktsiooni saab WiFi mooduli käivitada ja pääsupunktiga ühendada järgmiselt:

```

void Controller::setupWifi(String ssid, String password) {
  bitSet(MySignals.expanderState, EXP_ESP8266_POWER);
  MySignals.expanderWrite(MySignals.expanderState);

  MySignals.enableSensorUART(WIFI_ESP8266);
  delay(500);

  String cwjap = "AT+CWJAP=\"" + ssid + "\",\"" + password + "\"";

  sendData("AT", 1000, true);
  sendData("AT+CWMODE=1", 1000, true);
  if (sendData(cwjap.c_str(), 10000, true)) {
    tft.drawString("WiFi ok", 0, 30, 2);
  }
}
}

```

Kõigepealt on vaja WiFi mooduli tööle seadmiseks muuta vastava biti väärtus 1-ks, mille järel moodul käivitub. Seejärel tuleb seadistada UART pesasid kuulama just WiFi moodul.

Järgmiseks on kolm AT käsklust: “AT” kontrollib mooduli tööl olekut, “AT+CWMODE=1” seadistab mooduli pääsupunktiga ühendamiseks ja “AT+CWJAP”, millele on parameetriteks WiFi nimi ja salasõna, ühendab mooduli pääsupunktiga. Kui pääsupunktiga ühendumine õnnestub, kirjutatakse TFT ekraanile „WiFi ok“.

Peale mõõtmistulemuste lugemist sensorilt andmete saatmiseks veebirakendusse on kasutatud järgmist funktsiooni httpPost():

```
void Controller::httpPost(String& api, String& requestBody) {

    String request = "POST ";
    request += api;
    request += " HTTP/1.1\r\nHost: arduino-health-app.herokuapp.com\r\nContent-Type:
application/json\r\nContent-Length: ";
    request += requestBody.length();
    request += "\r\n\r\n";
    request += requestBody;

    String sendPostRequest = "AT+CIPSEND=";
    sendPostRequest += request.length();

    const char* cipStart = "AT+CIPSTART=\"TCP\", \"arduino-health-
app.herokuapp.com\", 80";

    sendData(cipStart, 2000, true);
    sendData(sendPostRequest.c_str(), 6000, true);
    sendData(request.c_str(), 4000, true);
    sendData("AT+CIPCLOSE", 1000, true);
}
```

Kõigepealt moodustatakse päringu sõne ja kahe AT käskluse sõne. Seejärel saadetakse käsklused järgmises järjekorras: „AT+CIPSTART“, mis loob TCP ühenduse defineeritud serveriga ja defineeritud pordil, „AT+CIPSEND“, mis saadab HTTP POST päringu ja mille parameetriteks on päringu tähemärkide arv ning seejärel päring ise. Lõpetuseks suletakse TCP ühendus „AT+CIPCLOSE“ käsklusega.

### 3.3 Bluetooth suhtlus

Bluetooth mooduli kontrollimine toimub MySignals'i teeki kasutades. Bluetooth'i ühenduse loomiseks kindla sensoriga on kõigepealt vaja sensori MAC aadressi. Kuna aadressi sensorite enda peale ei ole kirjutatud, siis on üks võimalus aadressi leidmiseks alla tõmmata mõni

vastava funktsiooniga rakendus nutitelefonile. iOS nutitelefonidel sobib selleks näiteks rakendus LightBlue, mis on vabalt allalaetav. Olles sensori sisse lülitanud on võimalik mobiilirakenduses ühendada nutitelefoni sensoriga ning on võimalik leida sensori MAC aadress. Allpool on toodud koodinäide, kuidas MySignals'i teeki kasutades Bluetooth'i moodul initsialiseerida ja sensoriga ühendada.

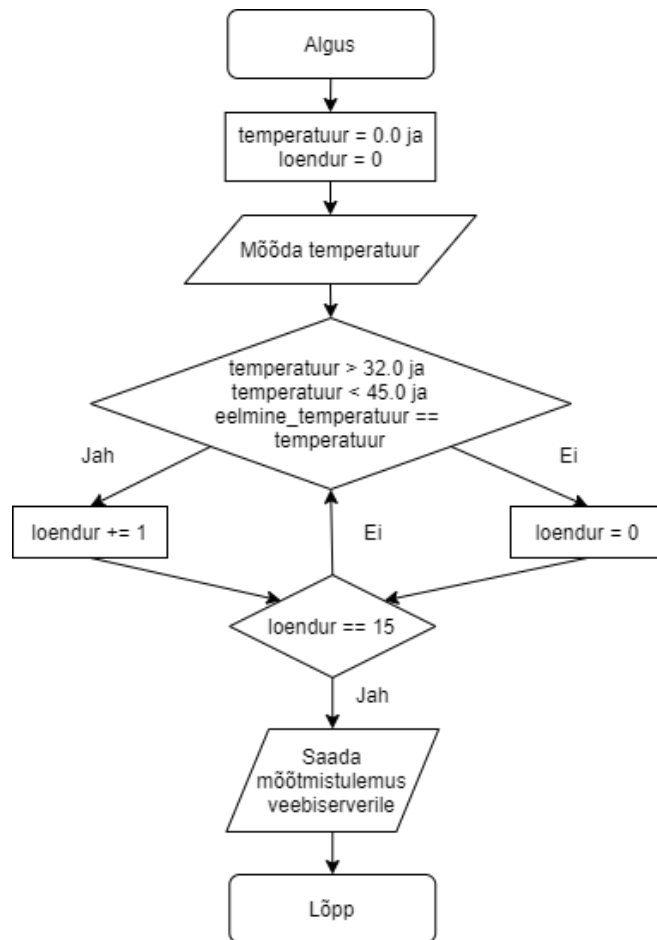
```
char MAC_SPO2[14] = "00A050042F13";
uint8_t available_spo2 = MySignals_BLE.scanDevice(MAC_SPO2, 1000, TX_POWER_MAX);
if (available_spo2 == 1) {
    if (MySignals_BLE.connectDirect(MAC_SPO2) == 1) {
        ...
        // Sensori mõõtmise initsialiseerimine ja vastuse lugemine
        ...
    }
}
```

### 3.4 Kehatemperatuuri mõõtmine

Kehatemperatuuri mõõtmise programm DS18B20 temperatuurisensoriga on vormistatud eraldi juhtprogrammina. Kehatemperatuuri mõõtja loodi, sest kehatemperatuur on osa töös kasutatavast varajase hoiatusskoori süsteemist. Lisaks on see näide sellest, et MySignals'i platvormile saab lisada ka teisi sensoreid, sest kehatemperatuuri mõõtja oleks saanud vormistada ka MySignals'i arendusplaadi külge. Kuna teine MySignals'i arendusplaat puudus, kasutati arendusplaati ESP32, mis sisaldab nii mikrokontrollerit kui ka WiFi moodulit. ESP32 arendusplaadile on võimalik programmikood laadida samuti Arduino arenduskeskkonnas.

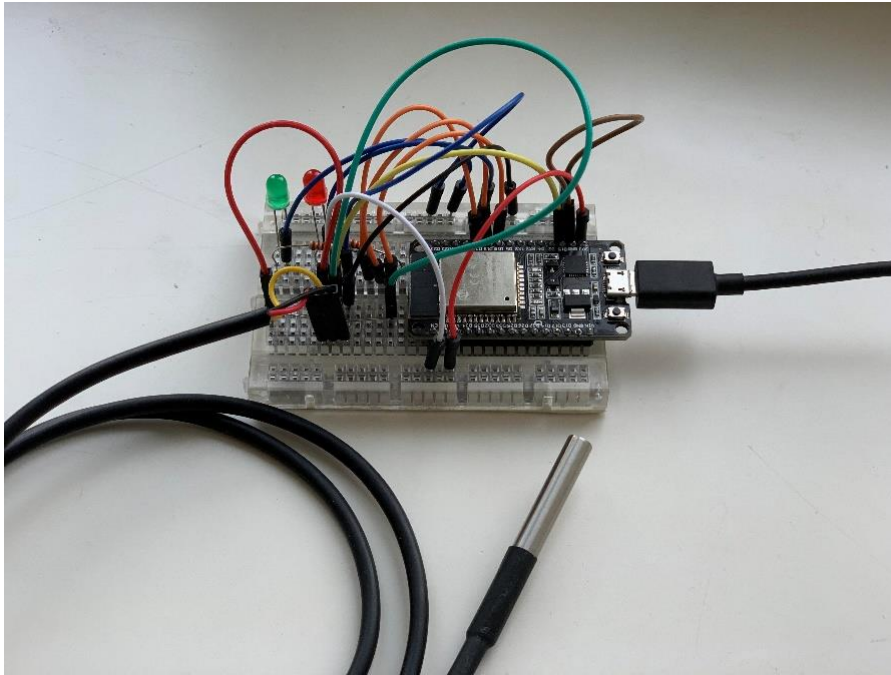
Programm kasutab sensoriga suhtlemiseks kahte teeki: OneWire ja DallasTemperature. WiFi pääsupunktiga ühendamiseks WiFi teeki ja HTTP päringu tegemiseks HTTPClient teeki. Nende teekide kasutamine lihtsustab programmeerija tööd, sest teeki kasutades ei ole vaja kasutada AT käsklusi ega kirjutada sõnena välja tervet HTTP päringut.

Selleks, et temperatuuri mõõtmine lõppeks ja teavitaks tulemusest, kui on saavutatud tegelik inimese kehatemperatuur on kasutatud lihtsat loenduriga algoritmi. Sensorilt küsitakse mõõtmistulemus iga sekund. Kui see on sama, mis eelmisel mõõtmisel, siis loendur kasvab ühe võrra ning vastasel korral läheb tagasi nulli. Juhul, kui loendur jõuab 15-ni mõõtmine lõpetatakse ning tulemus saadetakse veebirakendusse.



Joonis 6. Kehatemperatuuri mõõtmise algoritmi plokk skeem.

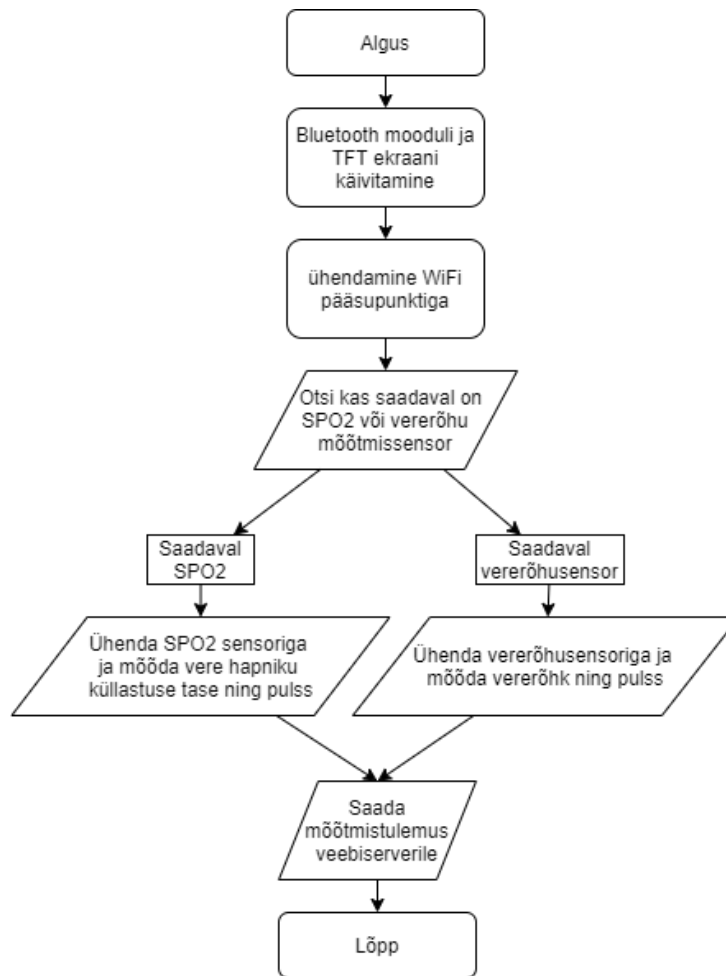
Kasutajale tagasiside andmiseks on kasutusel kaks LED tulukest. Mõõtmise jooksul vilguvad punane ja roheline kordamööda. Mõõtmise lõppedes jääb põlema roheline tuli, kui kasutaja kehatemperatuur on vahemikus 36 kuni 37 ning vastasel juhul jääb põlema punane tuli. Joonisel 7 on näha mikrokontrollerit ESP32 koos LED ja temperatuurisensori ühendustega.



Joonis 7. ESP32 koos LED ja temperatuurisensori ühendustega.

### 3.5 Loodud Arduino juhtprogramm

Arduino juhtprogramm ühendub Bluetooth abil vererõhusensori või SpO2 sensoriga ning teostab mõõtmise. Mõõtmistulemused kuvatakse TFT ekraanil ja saadetakse HTTP päringuna veebiserverile. Loodud Arduino programm töötab järgmiselt. Programmi `setup()` osas, mida käivitatakse ühe korra Arduino käivituses, käivitatakse kõigepealt Bluetooth moodul ning seejärel WiFi moodul, mis ühendatakse pääsupunktiga. Programmi `loop()` osas, kus olevat koodi jooksutatakse pidevalt uuesti, skaneeritakse, kas leidub defineeritud MAC aadressiga sensorit (SpO2 või vererõhusensor). Sensori leidmisel luuakse ühendus ning käivitatakse sensori mõõtmisprotsess, mille lõppedes saadab sensor tagasi mõõtmistulemused, Bluetooth ühendus katkestatakse. Mõlema sensori leidmisel luuakse ühendus SpO2 sensoriga. Mõõdetud tulemus saadetakse HTTP POST päringuna, eelpool kirjeldatud `httppost()` funktsiooniga, veebiserverile. Uue mõõtmise sooritamiseks on vaja sensor uuesti sisse lülitada. Joonisel 8 on kujutatud juhtprogrammi täitmise plokkskeem.



Joonis 8. Arduino juhtprogrammi plokkskeem.

Iga päringuga, mis saadetakse veebirakendusse, saadetakse lisaks mõõtmistulemusele ka programmi alguses määratud kasutajanimi. Kasutajanimi võimaldab veebirakenduses hoida erinevate kasutajate andmed lahus. Päringuga ei panda kaasa salasõna, sest veebirakendusega ei looda krüpteeritud HTTPS ühendust ning salasõna saatmine oleks ebaturvaline. Andmete edastuse ebaturvalisusest on autor teadlik ning rakendus sellisel kujul ei sobi reaalseks kasutamiseks.

HTTPS ühenduse loomine on ESP8266 mooduliga võimalik, kuid üsna aeglane ja võib mõne võtmepaari vahetamisel võtta mitu sekundit, kuid loodud rakenduse puhul ei oleks mõne sekundi ootamine takistuseks. Põhjus, miks HTTPS ühendust ei olnud võimalik kasutada, on selles, et krüpteeritud ühenduse kasutamine nõuab liiga palju Arduino mälu. Rakendus kasutaks umbes 5,6 kB ilma ühendust loomata ja iga ühendus vajaks veel 22 kB mälu. Seega ei ole Arduino Uno kasutades HTTPS ühendust võimalik luua [26].

### 3.6 Arendusprotsess

Selles alapeatükis kirjeldatakse kahte arendusprotsessi jooksul tekkinud probleemi ja nende lahendusi. Üks aspekt, mida peab MySignals'i arendusplaadi komponentide juhtimisel jälgima on see, et üle Serial pesade saadetud andmed jõuaksid õige komponendini ehk Bluetooth moodulini, WiFi moodulini või Serial monitori. Dokumentatsioonis ei olnud eraldi välja toodud, kuidas täpselt komponentide vahel käskude suunata. Arendusprotsessi alguses tekitas see autoris segadust, sest põhjustas esialgu programmi kummalist käitumist. Lahenduseks oli järgmiste funktsioonide kasutamine iga kord, kui muutus üle Serial pesade saadetavate andmete adressaat:

- `enableSensorUART(WIFI_ESP8266)` – Serial pesade andmed liiguvad WiFi mooduli ja mikrokontrolleri vahel
- `enableSensorUART(BLE)` – Serial pesade andmed liiguvad Bluetooth mooduli ja mikrokontrolleri vahel
- `enableMuxUART()` – Serial pesade andmed liiguvad ainult Arduino Serial monitori. Lõpetamiseks on funktsioon `disableMuxUART()`

Probleem, mis arendusprotsessi jooksul pidevalt esile kerkis, oli Arduino Uno väike mälu maht. Täpsemalt jäi puudu muutmälust, mida on Arduino Uno1 2048 baiti. Probleemi üheks lahenduseks oli, et funktsioonid kasutaksid parameetritena muutujate viitasid, mitte muutujate koopiasid. Teiseks saab Serial monitori väljundi kirjutamisel kasutada makrot `F()`. „Tere maailm!“ Serial monitori kirjutamiseks tuleb seda kasutada nii: `Serial.println(F(„Tere maailm!“))`. Nii hoitakse sõne „Tere maailm!“ Arduino püsimälu ning hoitakse kokku ruumi muutmälu. Kolmandaks, kuna mälu kasutus oli juhtprogrammis endiselt üle piiri ja programm jooksis selle tõttu kokku, oli vaja vähendada ka HTTP päringute kehade pikkuseid, muutes lühemaks JSON objekti võtmete nimesid.

Kokku võimaldasid muutmälu kokku hoidvad muudatused kasutusele võtta ka TFT ekraani, mille kasutamine muidu ei oleks olnud võimalik. Ekraanile kirjutatakse juhtprogrammis tervitussõnum ja teavitused Bluetooth mooduli tööle hakkamisest ning WiFi ühenduse loomisest. Mõõtmise lõppedes kuvatakse ekraanile sensori mõõtmistulemused.

### 3.7 Loodud veebirakendus

Loodud veebirakenduse eesmärgiks oli luua API lõpppunktid, kuhu Arduino saab teha HTTP POST päringuid sensori mõõtmistulemuste salvestamiseks ning kasutajaliides andmete visualiseerimiseks. Loodud API lõpppunktid on kirjeldatud tabelis 4.

Tabel 4. Loodud veebirakenduse API lõpppunktid.

API	Kirjeldus	Päringu keha näide
POST “api/bp/result”	Salvestab SPO2 sensori mõõtmistulemused	{s: 116, d: 70, p: 61}
POST “api/spo2/result”	Salvestab vererõhusensori mõõtmistulemused	{spo2: 98, pulse: 65}
POST “api/temperature/result”	Salvestab kehatemperatuuri mõõtmistulemused	{temperature: 36.6}
GET “api/get-filtered-results”	Võimaldab pärida filtreeritud tulemusi määrates sensori ja ajalise vahemiku	{from: 2020-04-13T15:17:03Z, to: 2020-04-15T15:17:03Z, sensor: „spo2“/„bp“/„temperature“}

Tavakasutajana on rakenduses näha kõiki kogutud andmeid. Sisse logituna näeb kasutaja enda andmeid. Antud vaates on kasutajal võimalik filtreerida oma terviseandmete mõõtmistulemusi valides soovitud sensorid ja kuupäevad. Vaade on näidatud joonisel 9. Teine vaade on “EWS Demo”, kus kasutaja viimaste mõõtmistulemuste põhjal on näha varajane hoiatusskoor, mis hindab kasutaja terviseseisundi tõsidust.

My Data EWS Demo Log out

Sensor:  SPO2  Bloodpressure  Temperature

Time:

Total measurements: 4

#	Systolic	Diastolic	Pulse	Time
1	112	83	66	13.4.2020 18:17:03
2	115	65	68	15.4.2020 14:45:03
3	115	65	68	26.4.2020 12:09:48
4	170	125	111	30.4.2020 20:48:36

Joonis 9. Loodud veebirakenduse kuvatõmmis.

Veebirakenduse realiseerimiseks kasutas autor serveri poolse rakenduse programmeerimiseks NodeJS'i ja kasutajapoolse rakenduse loomiseks VueJS raamistikku. Andmebaasina on kasutusel mitte-SQL andmebaas MongoDB. Need valikud tehti peamiselt põhjusel, et autor on nende tehnoloogiatega juba varasemalt tuttav ning need on ka head valikud väiksema rakenduse loomiseks.



Joonis 10. Mõõtmistulemus TFT ja nutitelefoni ekraanil.

Veebirakendus jookseb Heroku pilveteenuses aadressil <https://arduino-health-app.herokuapp.com/>. Joonisel 10 näeb tulemust peale mõõtmise lõppu. Mõõdetud SpO2 tulemus on näha nii MySignals arendusplaadi ekraanil kui ka veebirakenduses. Seega on näha kõik süsteemi komponendid sensorist veebirakenduseseni.

## 4. Kokkuvõte

Käesolevas töös anti ülevaade, mis on asjade internet ja kahe näite abil selgitati, kuidas on IoT lahendustest kasu tervishoius. Teiseks kirjeldati riistvara komponente, mida kasutati töös loodud terviseandmete mõõtmise ja jälgimise rakenduses. Viimases peatükis selgitati koodinäidete abil, kuidas kontrollida MySignals'i arendusplaadi Bluetooth ja WiFi komponente, kirjeldati kahte arendusprotsessi probleemi ning kahte loodud Arduino programmi ja veebirakendust. Sellega täidab viimane peatükk töö MySignals'i platvormi kasutamise näiteks olemise eesmärgi.

Töö käigus loodi kaks eraldi Arduino juhtprogrammi. Esimene programm ühendub SpO2 või vererõhusensoriga, teostab mõõtmise ja seejärel saadab saadud andmed loodud veebirakendusse, kus need salvestatakse. Teine programm mõõdab kehatemperatuuri ja saadab tulemuse veebirakendusse. Veebirakenduses on kasutajal võimalik näha enda mõõtmistulemusi ja kasutada varajast hoiatusskoori süsteemi, mis hindab kasutaja terviseseisundi tõsidust. Sellega täideti sissejuhatuses seatud rakenduse valmimise eesmärk.

Töö edasiarenduse üks võimalus on kasutada rohkem sensoreid ja teostada täpsemat ning personaalsemat andmete analüüsi. Rohkemate sensorite kasutamine vajaks ka erinevate stsenaariumite väljaarendamist, et vastata küsimusele, millal ja kuidas kasutajal loodud rakendusest kasu on. Lisaks võiks analüüside täpsemaks muutmisel kasutada andmeid patsiendi terviseandmete ajaloost.

Meditiiniliste asjade internet kasvab järgmistel aastatel kiirelt edasi ning leiab üha enam kasutust nii haiglates kui ka kodudes. Töös käsitletud MySignals'i riistvara platvorm on sobilik vahend patsiendi tervisenäitajate jälgimise rakenduste prototüüpimiseks, mille näiteks on töö käigus valminud prototüüplahendus.

## Viidatud kirjandus

- [1] Silcock D.J, Corfield A.R, Gowens P.A, Rooney K.D. Validation of the National Early Warning Score in the prehospital setting: 2015 [https://www.resuscitationjournal.com/article/S0300-9572\(15\)00011-8/fulltext](https://www.resuscitationjournal.com/article/S0300-9572(15)00011-8/fulltext) (15.04.2020)
- [2] Madakam S, Ramaswamy R, Tripathi S. Internet of Things (IoT): A Literature Review: 2015 [https://www.scirp.org/html/56616\\_56616.htm](https://www.scirp.org/html/56616_56616.htm) (25.04.2020)
- [3] International Data Corporation. The Growth in Connected IoT Devices Is Expected to Generate 79.4ZB of Data in 2025, According to a New IDC Forecast: 2019 <https://www.idc.com/getdoc.jsp?containerId=prUS45213219> (27.04.2020)
- [4] Haghi M, Thurow K, Habil I, Stoll R. Wearable Devices in Medical Internet of Things: Scientific Research and Commercially Available Devices: 2017 <https://synapse.koreamed.org/search.php?where=aview&id=10.4258/hir.2017.23.1.4&code=1088HIR&vmode=FULL> (27.04.2020)
- [5] Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications: 2015 <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7123563> (29.04.2020)
- [6] World Health Organization. A global brief on hypertension: 2013 [https://apps.who.int/iris/bitstream/handle/10665/79059/WHO\\_DCO\\_WHD\\_2013.2\\_eng.pdf;jsessionid=CBAA77FC0646D42DDBC0FD841B9AD179?sequence=1](https://apps.who.int/iris/bitstream/handle/10665/79059/WHO_DCO_WHD_2013.2_eng.pdf;jsessionid=CBAA77FC0646D42DDBC0FD841B9AD179?sequence=1) (22.04.2020)
- [7] Tartu Ülikooli Kliinikum. Vererõhu 24-tunni monitoring <https://www.kliinikum.ee/taastusravi/et/spordimeditsiin/teenused/2-uncategorised/100-vererohu-24-tunni-monitooring> (11.04.2020)
- [8] Ruiz-Fernández D, Marcos-Jorquera D, Gilart-Iglesias V, Vives-Boix V, Ramírez-Navarro Javier. Empowerment of Patients with Hypertension through BPM, IoT and Remote Sensing: 2017 <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5677452/#sec5-sensors-17-02273> (21.04.2020)
- [9] Ijaz A.U, Hashmat N, Shafiq F, Malik M.N, Khadija A. Early Warning Scoring System and in-hospital Mortality: 2009 [http://pjmhsonline.com/early\\_warning\\_scoring\\_system\\_and.htm](http://pjmhsonline.com/early_warning_scoring_system_and.htm) (21.04.2020)

- [10] Azimi I, Anzanpour A, Rahmani A.M, Liljeberg P. Self-aware Early Warning Score System for IoT-Based Personalized Healthcare : 2016 [https://www.researchgate.net/publication/311256501\\_Self-aware\\_Early\\_Warning\\_Score\\_System\\_for\\_IoT-Based\\_Personalized\\_Healthcare](https://www.researchgate.net/publication/311256501_Self-aware_Early_Warning_Score_System_for_IoT-Based_Personalized_Healthcare) (23.04.2020)
- [11] Doyle D.J. Clinical Early Warning Scores: New Clinical Tools in Evolution: 2018 [https://www.researchgate.net/publication/326737565\\_Clinical\\_Early\\_Warning\\_Scores\\_New\\_Clinical\\_Tools\\_in\\_Evolution](https://www.researchgate.net/publication/326737565_Clinical_Early_Warning_Scores_New_Clinical_Tools_in_Evolution) (21.04.2020)
- [12] Sparkfun. What is an Arduino? <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all> (03.04.2020)
- [13] Arduino. Arduino Uno Rev3. <https://store.arduino.cc/arduino-uno-rev3> (03.04.2020)
- [14] Tawil Y. Understanding Arduino UNO Hardware Design: All About Circuits, 2016 <https://www.allaboutcircuits.com/technical-articles/understanding-arduino-uno-hardware-design/> (03.04.2020)
- [15] Cooking hacks. MySignals HW v2 - eHealth and Medical IoT Development Platform for Arduino: 2017 <https://www.cooking-hacks.com/mysignals-hw-ehealth-medical-biometric-iot-platform-arduino-tutorial/index.html> (04.04.2020)
- [16] Libelium forum. ANDROID SDK-Test Connect: 2017 <https://www.libelium.com/forum/viewtopic.php?f=44&t=25259> (30.04.2020)
- [17] Fab Academy. ESP8266 Introduction. <http://fabacademy.org/archives/2015/doc/networking-esp8266.html> (27.04.2020)
- [18] Nettigo. <https://nettigo.eu/products/esp8266-01s-simple-easy-to-setup-wifi-connectivity-for-arduino> (27.04.2020)
- [19] RS. <https://uk.rs-online.com/web/p/bluetooth-modules/8977450/> (05.05.2020)
- [20] Techopedia. AT Command Set. <https://www.techopedia.com/definition/575/at-command-set> (08.04.2020)
- [21] Chen D. Wireless Communication with ESP8266: 2015 <http://fab.cba.mit.edu/classes/865.15/people/dan.chen/esp8266/> (08.04.2020)
- [22] Silicon Labs. BLE112. <https://www.silabs.com/wireless/bluetooth/bluegiga-low-energy-legacy-modules/device.ble112> (08.04.2020)
- [23] Berger A. Oscillatory Blood Pressure Monitoring Devices: 2001 <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1121444/> (06.04.2020)

- [24] Eesti Hüpertensiooni Ühingu ja Eesti Kardioloogide Seltsi arteriaalse hüpertensiooni juhised: 2004  
<http://www.ehy.ee/docs/pics/vererohk2004%20Juhised.pdf> (30.04.2020)
- [25] Last Minute Engineers. Interfacing DS18B20 1-Wire Digital Temperature Sensor with Arduino. <https://lastminuteengineers.com/ds18b20-arduino-tutorial/> (30.04.2020)
- [26] ESP8266 Arduino Core. BearSSL WiFi Classes. <https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/bearssl-client-secure-class.html> (01.05.2020)

## Lisad

### I. Rakenduse lähtekoodide repositooriumid

1. Loodud Arduino juhtprogrammi ja kehatemperatuuri mõõtmise lähtekoodi GitHub repositoorium: <https://github.com/karlmeldorf/Arduino-health-app>
2. Loodud veebirakenduse lähtekoodi GitHub repositoorium: <https://github.com/karlmeldorf/Arduino-health-app-web>

## II. Terminid

Arduino IDE Arduino integreeritud arenduskeskkond	Arduino IDE Arduino Integrated Development Environment
SRAM Staatiline juhupöördlusega mälu	SRAM Static random access memory
EEPROM Kustutatav programmeeritav püsिमälu	EEPROM Erasable Programmable Read Only Memory
TFT ekraan Kiletransistor ekraan	TFT screen thin-film transistor screen
SpO2 Pulssoksümeeter	SpO2 Pulse oximeter sensor
UART Asünkroonne jadavärati	UART Universal Asynchronous Receiver/Transmitter
Serial monitor Üle Serial pesade liikuvate andmete jälgija	Serial monitor Monitors the Serial port
Bit Informatsiooni ühik, mis saab olla 1 või 0	Bit Binary digit
Byte Informatsiooni ühik, mis on enamasti 8 biti pikk	Byte Unit of digital information that most commonly consists of 8 bits
MAC aadress Füüsilise võrguliidese unikaalne identifitseerija	MAC address Media access control address
HTTP Hüpertexti edastusprotokoll	HTTP HyperText Transfer Protocol

<b>HTTP POST</b> HTTP meetod andmete lisamiseks	<b>HTTP POST</b> HTTP method for sending data
<b>LED</b> Valgusdiod	<b>LED</b> Light-emitting diode
<b>JSON objekt</b> Andmevahetusvormingu objekt	<b>JSON object</b> JavaScript Object Notation object
<b>API</b> Rakenduse programmeerimisliides	<b>API</b> Application Programming Interface

### III. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Karl Meldorf**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **Terviseandmete jälgimine MySignals arenduskomplektiga**, mille juhendaja on Alo Peets, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Karl Meldorf

08.05.2020