TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Automaatikainstituut

ISS70LT

Vootele Vesterblom 111371IASMM

RASPBERRY PI-L PÕHINEV TEMPERATUURI MONITOORINGU SÜSTEEM

Magistritöö

Juhendajad: Ph.D. Eduard Petlenkov, dotsent
M.Sc. Aleksei Tepljakov

Tallinn 2013

Autorideklaratsioon

Olen koostanud antud töö iseseisvalt. Kõik töö kasutamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

Käesolevat tööd ei ole varem esitatud kaitsmisele kusagil mujal.

Autor: Vootele Vesterblom Allkiri:

Kuupäev:

Annotatsioon

Antud magistritöös arendatakse hiljuti turule tulnud mikroarvutile „Raspberry Pi“ temperatuuri monitooringu süsteemi.

Tarkvaraarenduse käigus tutvustatakse lugejale Raspberry Pi-d ja selle võimalusi.

Tarkvaraarenduse käiku kirjeldades tuuakse välja tarkvaraarendamise põhimõtted ning prototüübid, mille abil on tarkvaranõudeid lihtsam kirjeldada, sellele lisaks antakse lühiülevaade Raspberry Pi peal töötavatest kasutajaliidese raamistikest.

Tarkvaralahenduse kirjelduses tutvustatakse lahenduse väljatöötamise etappe, tarkvara struktuuri ning selle implementatsiooni. Lisaks tutvustatakse antud töös valminud tarkvara kasutajaliidest.

Töö lõpus on välja toodud töö käigus arendatud tarkvaralahenduse edasiarendamise võimalused.

Magistritöö on kirjutatud eesti keeles ning sisaldab teksti 72 leheküljel, 10 peatükki, 16 joonist, 1 tabel.

Abstract

This thesis presents to the reader a development of a temperature monitoring system on a microcomputer named „Raspberry Pi“, that recently came onto the market.

The user is presented with an introduction of Raspberry Pi and its capabilities.

During the development process the user will be presented with software developement principles and prototypes, which help facilitate the application development process by making the part of writing software requirements easier.

During the completion of the software the user is presented with the development phase, the software structure and its implementation. In addition to this the user will be presented with an introduction to the software’s graphical user interface.

At the end of the paper some guidelines of where to go next are outlined.

The thesis is in Estonian and contains 72 pages of text, 10 chapters, 16 figures, 1 table.

Lühendite ja mõistete sõnastik

apt-get	Tarkvara pakihaldus programm Linux tüüpi operatsioonisüsteemides.
BBC Micro	Mikroarvutide seeria, mida toodeti 1980-ndate alguses, et õpilastele arvuteid ja arvutiteadust õpetada.
Ethernet	Juhtmega kohtvõrgu tehnoloogia, mis on mõeldud kohtvõrkudes kasutamiseks.
FTP	*File Transfer Protocol*, failiedastusprotokoll, mis on ettenähtud failide edastamiseks arvutite vahel kasutades interneti ühendust.
FTPS	*FTP Secure*, turvalisust pakkuv laiendus tavalisele failiedastusprotokollile. FTPS lisab TSL ja SSL toe.
GPIO	*General Purpose Input/Output*, üldiseks kasutamiseks mõeldud kontaktide kogum kiibil, millel puudub kindel otstarve ja mille otstarvet saab programmeerida kiibi töötamise ajal.
GUI	*Graphical User Interface*, graafiline kasutajaliides.
HDMI	*High-Definition Multimedia Interface*, kõrglahutusega multimeedia kasutajaliides on standard kõrge kvaliteediga heli ja pildi edastamiseks.
Hot-swap	Käigultvahetus, komponendi eemaldamine arvutisüsteemist ja uue juurde lisamine ilma arvutisüsteemi töö seiskamiseta.
IEC	*International Electrotechnical Commission*
IEEE	*Institute of Electrical and Electronics Engineers*
IP aadress	*Internet Protocol* aadress, internetiaadress.
ISO	*International Organization for Standardization*, Rahvusvaheline standardiorganisatsioon.
Linux	Unixi laadne operatsioonisüsteem, mis on enamasti vabavaraline.
Makefile	Failid, mis spetsifitseerivad kuidas ja mis järjekorras peab erinevaid lähtekoodi faile kompileerima ja kokku liitma, et nendest rakendus kokku panna.
MATLAB	*Matrix Laboratory*, tarkvarakeskkond, mis on mõeldud tehniliste arvutuste jaoks ja mis pakub erinevaid mudelite koostamise ja analüüsi võimalusi.
Micro-USB	USB konnektor.

MP3	*MPEG-1 audio layer 3,* digitaalsete helifailide tihendamise algoritm.
Octave	Kõrgtaseme programmeerimiskeel, mis on peamiselt mõeldud numbriliste arvutuste jaoks. Octave sarnaneb MATLABile.
OpenGL ES	*OpenGL for Embedded Systems,* 3-D graafikakeel, mis on mõeldud sardsüsteemide jaoks.
PID kontroller	Proportsionaal-integraal-diferentsiaal kontroller.
Plugin	Tarkvaramoodul, mis lisab tarkvarasüsteemile teatud omadusi või teenuseid.
qmake	Programm, mis automatiseerib Makefile-de genereerimist. Mõeldud peamiselt QT raamistikku kasutavate projektide jaoks.
RCA connector	RCA pistikühendus on mõeldud audio- ja videosignaalide edasikandmiseks.
RJ45	*Registered jack 45,* RJ45 on konnektor, mida kasutatakse Ethernet võrgukaablites.
RPC	*Remote procedure call,* kaugprotseduurikutse on protokoll, mis võimaldab ühel arvutil asuval programmil käivitada teisel arvutil asuvat programmi ilma spetsiaalseid protseduure kasutamata.
RS-232	*Recommended standard-232,* järjestikliidese standard.
RTC	*Real-time clock,* reaalaja kell arvutitel, mis peab arvet käesoleva aja kohta.
RxD/TxD	*Receive data/transmit data, a*ndmeliinid järjestikliidestel.
SCP	*Secure copy,* protokoll andmete turvaliseks edastamiseks kahe arvuti vahel.
SD/MMC	*Secure Digital/MultiMediaCard,* mälukaardi formaadid, mida kasutatakse enamasti kaasaskantavates seadmetes näiteks mobiiltelefonides, digitaalkaamerates, MP3-mängijates.
SDRAM	*Synchronous dynamic random access memory,* dünaamiline suvapöördusega mälu, mis on sünkroniseeritud keskprotsessori siiniga.
SFTP	*SSH File Transfer Protocol/Secure File Transfer Protocol,* turvaline failiedastusprotokoll.
SMTP	*Simple Mail Transfer Protocol,* lihtne meiliedastusprotokoll, protokoll e-kirjade saatmiseks.
SoC	*System on a chip,* süsteemikiip. Ühte kiipi mahutatud terviklik toode

SSH	*Secure Socket Shell,* turvaline kest, UNIXi põhine käsuliides ja protokoll, mis võimaldab turvalist sisselogimist kaugarvutisse.
SSL	*Secure Sockets Layer,* turvasoklite kiht. Infoturbe protokoll üle interneti edastavate andmete turvalisuse tagamiseks.
TLS	*Transport Layer Security,* transpordikihi turbeprotokoll. Protokoll, mis võimaldab klient-server rakendustel omavahel turvaliselt suhelda üle interneti.
UART	*Universal asynchronous receiver/transmitter,* universaalne asünkroontransiiver, mikrokiip, mis juhib arvutit ja välisseadmeid ühendavat jadaliidest.
UI	*User Interface,* kasutajaliides.
USB	*Universal Serial Bus,* universaalne järjestikliides. USB lubab käigultvahetamist.
Widget	Vidin, kasutajaliidese element, mis kuvab informatsiooni või pakub kasutajale võimalust suhelda operatsioonisüsteemi või programmiga.

Jooniste nimekiri

Joonis 1. Raspberry Pi trendimine Google Trendis. 15
Joonis 2. Ülevaade Raspberry Pi-st koos sisendite ja väljunditega. 16
Joonis 3. GPIO päise kontaktide loetelu ja kontaktide nimetused. 21
Joonis 4. LogTag Analyzer – Graafiku temperatuur/aeg kuva. 24
Joonis 5. Antud töös kasutatud temperatuuriandur. 40
Joonis 6. Klassidiagramm töös kasutatud klassidest. 51
Joonis 7. Voodiagramm: Programmi käivitamine. 52
Joonis 8. Voodiagramm: Mõõtmiste tegemise käivitamine. 53
Joonis 9. Jadaskeem: Andurilt temperatuuri andmete saamine ja taimeri intervallid. 54
Joonis 10. Voodiagramm: Mõõtetulemuste salvestamine ja graafikute joonistamine. 55
Joonis 11. Temperatuuri monitooringu tarkvara kontrollpaneel. 58
Joonis 12. Peaakna temperatuurigraafik. 59
Joonis 13. Järjestikliidese seadete dialoog. 60
Joonis 14. E-Maili seadete dialoog. 61
Joonis 15. Alarmide dialoog. 62
Joonis 16. E-Mailile saadetud päevase statistika graafik. 63

Tabelite nimekiri

Tabel 1. Raspberry Pi süsteemkiibi ülevaade. *17*

Sisukord

1. SISSEJUHATUS 12

2. RASPBERRY PI 14

2.1. Raspberry Pi saamislugu 14
2.2. Ülevaade Raspberry Pi-st 16
2.3. Täpsemalt Raspberry Pi sisendite ja väljundite võimalustest 19

3. NÄITEID 23

3.1. Mis asi on temperatuuri logger 23
3.2. LogTag tarkvara 24
3.3. Fluke Calibration - LogWare III 25
3.4. OMEGA® iSD-TC 27
3.5. Tähtsamad võimalused mida pakutakse kommertsiaalsetes tarkvaralahendustes 28

4. NÕUDED ARENDATAVALE TARKVARALAHENDUSELE 29

4.1. Programmi nõuete kirjeldamisest 29
4.2. Nõuded Raspberry Pi-l põhinevale temperatuuri monitooringu programmile 31
4.3. Kasutajaliidese disain 35

5. VÕIMALIKUD LAHENDUSED RASPBERRY PI PEAL TEMPERATUURI MONITOORINGU TARKVARA ARENDAMISEKS 36

5.1. Nõuete analüüs 36
5.2. Programmeerimiskeeled ja raamistikud 37
5.3. Graafilise raamistiku valimine 37
5.4. Sobiva sisendi valimine temperatuuri anduri jaoks 38
5.5. Valikute kokkuvõte 40

6. TARKVARALAHENDUSE IMPLEMENTATSIOON 41

6.1. Raspberry Pi operatsioonisüsteem 41
6.2. QT tutvustus 41
6.3. Koodi taaskasutamise põhjused 43
6.4. Väliste teekide kasutamine antud töös 44
6.5. Kokkuvõte teekide kasutamisest antud töös 50

7. TARKVARALAHENDUSE STRUKTUUR 51

7.1. Klasside tutvustus 51
7.2. Programmi töö diagrammid 52
7.3. Programmi lähtekoodi failide tutvustus 55

8. TARKVARALAHENDUSE TUTVUSTUS 57

8.1. Programmi transportimine Raspberry Pi peale ja selle kompileerimine 57
8.2. Kasutajaliidese peaaken 58
8.3. Konfiguratsiooni dialoogide tutvustus 60
8.4. Näide programmi poolt saadetavast e-mailist 62

9. EDASIARENDAMISE VÕIMALUSED 64

9.1. HeaterLabKitiga integreerimine 64
9.2. Lisa graafikute lisamine .m failide abil 64
9.3. Anduri temperatuuri võrdlemine meteoroloogiliste andmetega 65
9.4. Veebiliidese kaudu anduri(te)lt info kätte saamine 66

10. KOKKUVÕTE 68

KASUTATUD KIRJANDUS 69

1. Sissejuhatus

Käes on 21. sajand ja statistikaameti andmetel on Eestis vähemalt 75% arvutikasutajaid: „*2012. aasta I kvartalis oli 75%-l leibkondadest kodus internetiühendus, teatab Statistikaamet. Eelmise aasta sama ajaga võrreldes on selliste leibkondade osatähtsus suurenenud neli protsendipunkti.*“ [1].

Selline hulk kasutajaid (eeldades, et internetiühenduse olemasolu korral seda kasutatakse) peaks tähendama ka seda, et Eestis on IT valdkond populaarne ja tööjõupuudust seal ei ole, aga tundub, et see nii ei ole. Miks muidu annab Infotehnoloogia Liit teada, et Eestis on IT-spetsialistide puudus [2].

Selline hulk kasutajaid ja selline tööjõupuudus peamiselt arvutitega tegelevas valdkonnas kätkevad endas teatud vastuolu. Kuskil on midagi valesti, sest tundub, et IT valdkonnas pole juurdekasv nii suur kui tahetakse, mis näitab seda, et noortel pole tahtmist seda õppida. Õppimise tahe tuleb aga headest õpetajatest ja headest õppematerjalidest.

„Õnneks“ olid sama probleemi ees ka antud töös tutvustatava mikroarvuti Raspberry Pi loojad, kes samuti nentisid fakti, et on langemas nende noorte hulk, kes tahavad südamega arvutiteadust õppida.

Probleemi lahendamiseks alustasid nad tööd riistvaralise õppematerjali tegemiseks, millest lõpus kasvas välja Raspberry Pi ja antud õppevahendi kasutamist antud töös tutvustatakse.

Antud töö põhiline eesmärk on temperatuuri monitooringu tarkvara arendamine Raspberry Pi peal. Tarkvaraarenduse käigus tutvustatakse lugejale ka Raspberry Pi-d ja tarkvaraarenduse võimalusi selle peal. Tutvustuse käigus peaks lugeja saama selgema ülevaate tarkvaraarendusest Raspberry Pi peal.

Antud töös tutvustatakse lugejale kõigepealt Raspberry Pi-d, täpsemalt kirjeldatakse kuidas tekkis idee selle loomiseks. Sellele lisaks tuuakse ära millised sisendid ja väljundid Raspberry Pi-l on ning seejärel tehakse tähtsamatest sisenditest ja väljunditest detailsem ülevaade.

Peale sissejuhatust Raspberry Pi-sse tutvustatakse kommertsiaalseid temperatuuri monitooringu tarkvaralahendusi selleks, et saada ülevaade/aimdus milline võiks antud töö käigus arendatav tarkvaralahendus välja näha.

Peale kommertsiaalsete tarkvaralahenduste tutvustust pannakse paika arendatava tarkvaralahenduse nõuded selleks, et tarkvara arendades oleks mingi kindel eesmärk, mille poole püüelda.

Peale nõuete spetsifitseerimist analüüsitakse võimalikke tarkvaralisi vahendeid, millega saab Raspberry Pi peal täita eelnevas osas spetsifitseeritud nõudeid. Analüüsi tulemusena valitakse välja ka kindlad vahendid, mille abil hakatakse tarkvaralahendust arendama. Lisaks valitakse antud osas välja ka Raspberry Pi sisendid, mille abil tarkvaralahendus hakkab suhtlema temperatuurianduriga.

Pärast vahendite valikut kirjeldatakse tarkvaralahenduse implementatsiooni Täpsemalt kirjeldatakse seda kuidas valitud vahendeid kasutati ja lisaks põhjendatakse miks antud töös kasutatakse tarkvara taaskasutamise printsiipi.

Peale implementatsiooni kirjeldust jõuab järg programmi struktuuri kirjelduseni, kus tutvustatakse töö käigus valminud rakenduse klasside struktuuri ja tutvustatakse lähemalt tähtsamaid tegevusi. Lisaks kirjeldatakse lühidalt, mis on erinevate lähtekoodi failide otstarve antud tarkvaralahenduses.

Struktuuri kirjeldamise järel tutvustatakse kasutajale tarkvara kasutajaliidest kasutades selleks visuaalseid objekte. Lisaks tutvustatakse antud töö käigus kasutatud temperatuuriandurit.

Kõige lõpuks tuuakse välja mõned edasiarenduse võimalused, mille suunas võiks antud tarkvara edasi arendada.

2. Raspberry Pi

2.1. Raspberry Pi saamislugu

Et aru saada, mis on Raspberry Pi, tuleb kõigepealt rääkida selle loomise tagamaadest.

Raspberry Pi loomise idee tekkis sellepärast, et Cambridge-i ülikooli õppejõud nentisid aastal 2006 fakti, et aasta aastalt langeb arvutiteaduses õppivate tudengite arv ja tase. Otsas olid need ajad, kus arvutiteadust tulid peamiselt õppima lapsed/noored, kes olid noorena omal käel programmeerimist õppinud, arvuteid lahti võtnud (nö. tõelised insenerid), vaid alanud olid ajad, kus infotehnoloogia oli muutumas popiks valdkonnaks, mida tahtsid kõik õppida, ka need, kes olid parimal juhul ainult oma veebilehte koostanud [3].

Alanud olid mingis mõttes muret tekitavad ajad.

Ideeks oli luua midagi sellist, mida poleks raske kasutada, aga mida kasutades lapsed leiaksid ajendi, mis tõukaks neid proovima arvutitega programmeerima.

Eesmärgiks oli tollal kehtivatesse õppekavadesse, kus õpetati pigem arvuti ja selle programmide käsitlemist, kui nende loomist, tuua sisse tarkvara loomise osa [4].

Raspberry Pi-ga taheti luua samasugust tungi arvutiteaduse õppimisele nagu seda tegi 30 aastat tagasi BBC Micro [3], [5], [6].

Võimalus väikse ja tarviliku arvuti loomiseks oli olemas, sest võimsad ja väikesed protsessorid (mobiilsed protsessorid) olid juba olemas ja nende hind oli odavnemas. Lisaks olid nad piisavalt võimsad, et pakkuda head multimeedia võimekust – mis peaks eelkõige meeldima lastele, keda nähakse selle mikroarvuti peamise kasutaja sihtgrupina [3].

Peale Raspberry Pi müüki tulekut arvati, et see muudab arvutiteaduse õpetamist:

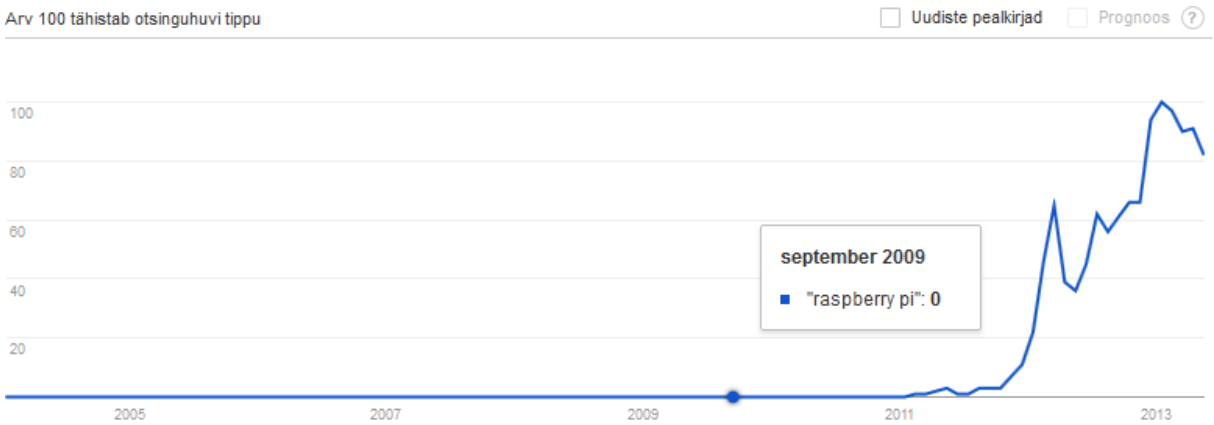
Gareth Mitchell arvab, et Raspberry Pi on sellepärast edukas, et ta maksab kõigest 25$ ja seetõttu on selle võimaluse (Raspberry Pi abil arvutiteadust õpetada) kasutamine suurenenud ka arengumaades, kus on veel avastamata noori helgeid päid [7].

Aga samas oli ka kahtlejaid:

Piers Bizony näiteks arvab, et arvutiteaduse õpetamine ei saa tulevikus sõltuda ainult ühest tarbijaproduktist, mis on lihtsalt mingi komponentide kogum [8].

Lisaks arvab ta, et Raspberry Pi ei too arvutiteaduse juurde uusi noori inimesi, sest Raspberry Pi-d hakkavad enamasti kasutama noored, kellele niikuinii meeldib juba programmeerida. Veel arvab ta, et Raspberry Pi reklaamimine kui 25$ arvutina on eksitav, sest sellele on juurde vaja osta lisaseadmed (monitor, hiir, klaviatuur ja salvestamise seade).

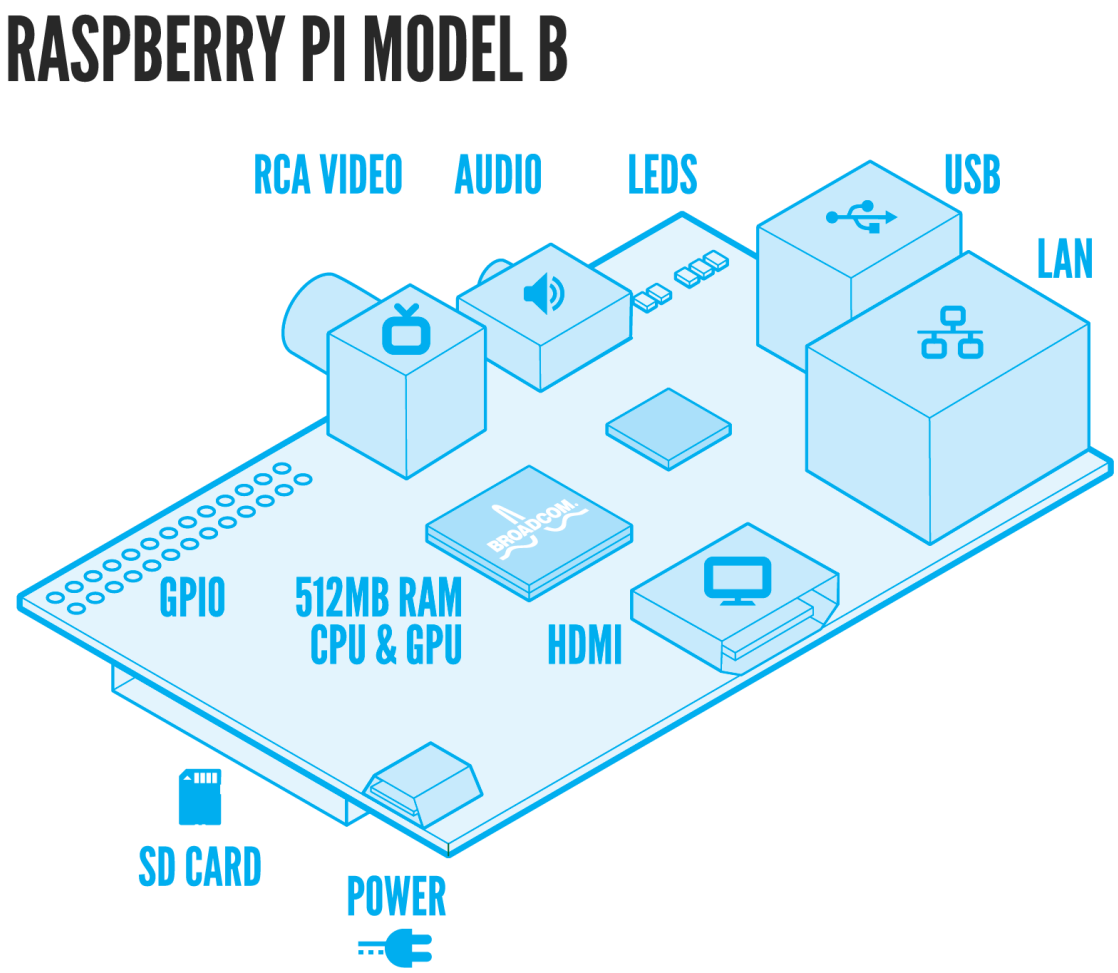
Kuid kui vaadata „Raspberry Pi“ trendimist Google Trendis [9]:



Joonis 1. Raspberry Pi trendimine Google Trendis.

Ning lisaks veel seda, et peale ülemaailmset müüki tulekut aprillis 2012 [10] jõuti aasta hiljem miljoni müüdud ühikuni [11] võib öelda, et skeptikute juttu pole keegi veel tõsiselt võtnud.

2.2. Ülevaade Raspberry Pi-st



Joonis 2. Ülevaade Raspberry Pi-st koos sisendite ja väljunditega.

Järgnevad spetsifikatsioonid pärinevad Raspberry Pi korduvalt küsitud küsimuste lehelt [12].

1. **Suurus**

   Mõõtmed - 85.60mm x 56mm x 21mm (pikkus x laius x kõrgus).

   Kaal – 45g.

2. System on a chip (SoC)

Tabel Raspberry Pi-s kasutatavast SoC-st:

Tabel 1. Raspberry Pi süsteemkiibi ülevaade.

<table>
<tr><td colspan="2">System on a Chip
BroadCom BCM2835 [13]
BroadCom BCM 2835 on täiskõrglahutusega multimeediaprotsessor, mis on mõeldud kõrget multimeediajõudlust nõudvatele sard- ja mobiilrakendustele ja mis on disainitud ja optimiseeritud saavutamaks võimalikult head kasutegurit.
BCM 2835 kasutab Broadcom Videocore 4 tehnoloogiat, et võimaldada multimeediarakenduste kasutamist.
Antud kiip tarbib vähe voolu ja on kõrge jõudlusega ja seetõttu sobib ta kasutamiseks mobiilsetes vahendites, mis on enamjaolt akutoitel.
Teiseks sobib ta multimeedia seadmete jaoks, mille puhul on vaja, et seade ei tekitaks taustamüra nagu näiteks erinevad DVD/Blu-Ray mängijad, mängukonsoolid, televiisorid.</td></tr>
<tr><td>CPU (protsessor)

700MHz ARM11 ARM1176JFZ-S [14]
Protsessoriks on ARM11 protsessor, mis on kiirendatud 700MHz-ni ja mis kasutab ARMv6 arhitektuuri.
ARM11 protsessoreid kasutatakse nutitelefonides, digitaal TV-des, e-lugerites ja paljudes muudes rakendustes, kus on vaja väikest, aga võimekat protsessorit.</td><td>GPU (videoprotsessor)

Broadcom Videocore 4
Videocore 4 on madala võimsuse ja kõrge jõudlusega videoprotsessor.
Videoprotsessor pakub OpenGL ES 1.1/2.0 võimalust. Videocore 4 võimaldab kodeerida/dekodeerida täiskõrglahutusega 1080p videot.
Oma 1080p võimaluse poolest sobib Videocore 4 multimeedia seadmete jaoks.</td></tr>
</table>

3. Mälu (SDRAM)

Mudel A – 256MiB

Mudel B – 512MiB

4. USB 2.0 pesad

Mudel A – 1.

Mudel B – 2.

5. **Videoväljundid**

Komposiitvideo (RCA).

HDMI.

PS: Videoväljundeid ei saa korraga kasutada.

6. **Audioväljundid**

3.5mm väljund.

HDMI.

7. **Audiosisendid**

Audiosisendeid ei ole, aga vajadusel saab nende jaoks lisada USB seadme.

8. **Salvestusvõimalused**

Kõvaketas puudub.

Salvestusseadmena saab kasutada SD/MMC kaarti.

9. **Emaplaadil olev võrgutugi**

Mudel A – Puudub.

Mudel B – 10/100 juhtmega Ethernet RJ45.

10. **Madalatasemelised lisaseadmed**

GPIO kontaktid, mis pole kasutajale kergesti kättesaadavad.

11. **Reaalaja kell**

Raspberry Pi-l puudub reaalaja kell (RTC), aga vajadusel saab selle lisada GPIO kontaktide külge.

12. **Voolutarve**

Mudel A – 500mA, (2.5W).

Mudel B – 700mA, (3.5W).

13. Jõuallikas

5V alalisvool (DC) Micro-USB pessa ühendatud kaabli abil või GPIO kontaktide kaudu.

14. Tarkvara

Operatsioonisüsteem – Raspberry Pi toetab Linuxil põhinevaid operatsioonisüsteeme.

2.3. Täpsemalt Raspberry Pi sisendite ja väljundite võimalustest

Antud osas tutvustatakse täpsemalt Raspberry Pi-l olevaid tähtsamaid sisendeid ja väljundeid.

1. USB

Raspberry Pi- l on USB 2.0 pesad.

USB loodi selleks, et standardiseerida välisseadmete ühendust arvutiga. USB abil saab arvuti külge ühendada peaaegu igasuguseid seadmeid (arvutihiir, klaviatuur, mobiiltelefon, modem, monitor, skänner, digikaamera, MP3-mängija, väline kõvaketas, traadita võrgukaart jne.). USB seadmete ühendamiseks ei ole vaja arvutit algkäivitada [15].

Põhiliselt ühendatakse Raspberry Pi külge USB pesasse klaviatuure, hiiri, väliseid kõvakettaid kui ka USB hub-e, et siis selle külge lisaseadmeid ühendada.

Lisaks saab juhtmeta võrgu saamiseks USB pessa ühendada USB Wi-Fi adaptereid ja paljusid teisi seadmeid.

2. HDMI

HDMI – video väljund, audio väljund.

HDMI abil saab edastada videopilti koos heliga [16].

Raspberry Pi-l saab HDMI pesa kasutada Raspberry Pi ühendamiseks kas monitori või teleka külge.

3. Ethernet

Ethernet – Juhtmega kohtvõrgu tehnoloogia IEEE 802.3 [17].

Raspberry Pi-l on Etherneti kasutamiseks RJ-45 pistik.

Ethernet võimaldab Raspberry Pi-l ühendust internetiga.

4. GPIO päis

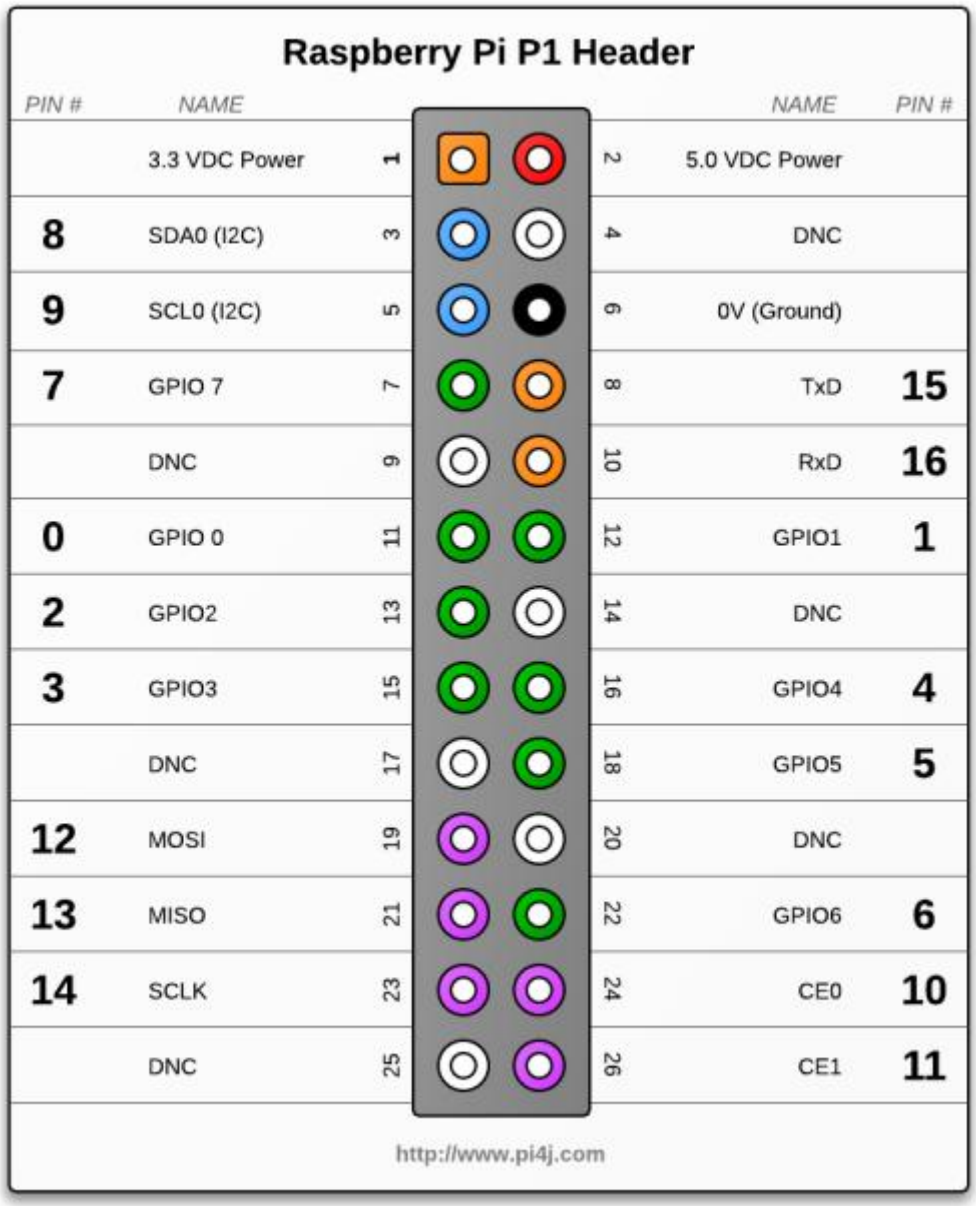
GPIO (General purpose input/output) – üldine sisend/väljund kontaktide kogum kiibi peal, mida saab kontrollida (programmeerida) kasutaja poolt kiibi töötamise ajal [18].

GPIO päisel ei ole kindlat otstarvet.

Raspberry Pi-l on 26 GPIO kontakti [19].

Hoiatusena tuleb öelda nii palju, et kuna Raspberry Pi-l ei ole ülepinge kaitset, siis ei ole soovitatav GPIO kontakte otse kasutada vaid selleks on soovitatud kasutada lisaplaati, millel on olemas puhvrid, mis siis kaitsevad kiipi.

Järgnevalt joonis [20] Raspberry Pi GPIO ühenduste päise kohta:



Joonis 3. GPIO päise kontaktide loetelu ja kontaktide nimetused.

GPIO päis omab kontakte järgmiste ühenduste jaoks [21]:

SPI (*Serial Pheripheral Interface Bus*)

Pildil kontaktid nimedega: MOSI, MISO, SCLK, CE0, CE1.

Andmesidesiini protokoll mitme seadme vaheliseks sünkroonseks täisdupleks ühenduseks. Seadmed kommunikeeruvad ülem-alluv-süsteemis (master/slave) [22].

Raspberry Pi puhul on Raspberry Pi ülemaks, kes andmesidet juhib.

I^2C (*Inter Integrated Circuits*)

Joonisel 3 kontaktid nimedega: SDA0 ja SCL0.

80-ndatel Philipsi poolt loodud järjestiksiin, mida kasutati lihtsaks kommunikatsiooniks komponentide vahel, mis asusid ühe skeemi peal. I^2C on samuti ülem-alluv süsteemi kasutav protokoll [23].

I^2C puhul on Raspberry Pi puhul Rasperry Pi ülemaks, kes andmesidet juhib.

UART (*Universal asynchronous receiver/transmitter*) [24]

Joonisel 3 kontaktid nimedega: TxD ja RxD.

UART puhul on tegu universaalse asünkroontransiiveriga [25] teisisõnu järjestikliidesega (RS-232).

UART abil saab Raspberry Pi suhelda teiste seadmetega, kes omavad sama ühendusvõimalust.

Kui SPI ja I^2C puhul on tegu süsteemisiseste kommunikatsiooniprotokollidega, siis järjestikliidest kasutatakse süsteemiväliste seadmetega suhtlemiseks, milleks võivad olla erinevad mikrokontrollerid, mis omavad järjestikliidese pesa.

Peale nende kontaktide jääb GPIO-l üle veel 8 pinni, mida saab kasutada üldiste sisend/väljund kontaktidena.

3. Näiteid

Selleks, et paremini mõista milline temperatuuri jälgimise süsteem peab välja nägema on vaja analüüsida, millised kommertsiaalsed tarkvaralahendused välja näevad ja missuguseid võimalusi nad kasutajale pakuvad.

Töö hilisemates osades, kus tutvustatakse töö käigus väljaarendatud tarkvaralist lahendust, tuuakse tarkvaralahenduse võimalusi tutvustades välja ka sarnasused antud peatükis nimetatutega.

Järgnevates alam-peatükkides tuuakse välja kolme temperatuurimonitooringu tarkvara tähtsamad omadused, mille põhjal saab hilisemates osades otsustada, millisena võiks antud töö käigus välja arendatud tarkvaralahendus välja näha.

3.1. Mis asi on temperatuuri logger

Temperatuuri logger on väike temperatuuri mõõtev ja salvestav seadeldis, mis suudab mõõta temperatuuri etteantud intervalli tagant. Temperatuuri loggereid kasutatakse enamasti saadetiste temperatuuri jälgimiseks või siis temperatuuri jälgimiseks erinevates keskkonnatingimustes [26][27].

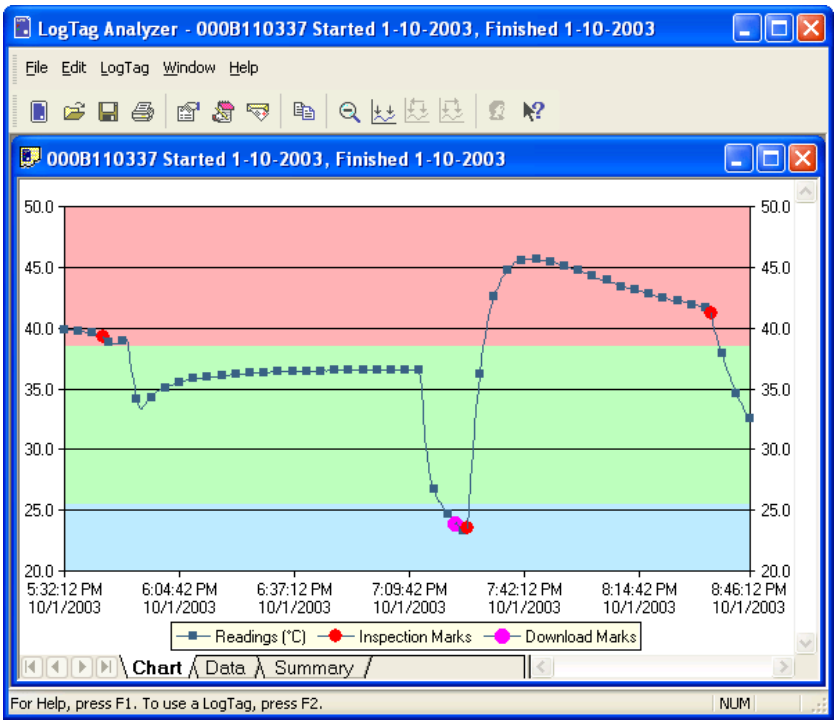
Enamasti kasutatakse temperatuuri loggereid arvutitest eraldi ja hiljem kantakse nende pealt andmed üle arvutisse, kus neid analüüsitakse.

Kuna on vaja üldpilti temperatuuri andmeid analüüsivatest seadmetest, siis on siin töös vaadatud ka näiteid temperatuuri loggerite andmete analüüsimiseks mõeldud tarkvaradest, mis analüüsi reaalajas ei tee, aga mille analüüsi võimalusi oleks tarvilik juurutada reaalajas temperatuuri mõõtmisi ja analüüsi tegevas programmis.

3.2. LogTag tarkvara

LogTag Analyzer [28]

Järgnevalt näide LogTag Analyzeri temperatuur/aeg graafikust [28]:



Joonis 4. LogTag Analyzer – Graafiku temperatuur/aeg kuva.

Võimalused:

Graafiku kuvamine – Antud tarkvara suudab genereerida mõõteandmetest temperatuur/aeg graafiku, mille peal on eraldi ära märgitud andmepunktid, mille ajal temperatuur oli üle või alla normaalvahemiku.

Graafikul teatud piirkonna suurendamine – Graafikut on võimalik teatud piirkonda hiirega valides suurendada.

Raporti tegemine

Tähtsamad osad, mis raportis välja tuuakse:

- Mõõteintervall.
- Alarmide olemasolu.
- Alarmide nõuded.

    - Alumine ja ülemine piir.
    - Mitu järjestikust korda peab alarm üle piiri olema.
- Andmete mõõtmise alustamise ja lõpetamise aeg.
- Temperatuuri statistika.
    - Kõige madalam temperatuur.
    - Kõige kõrgem temperatuur.
    - Eksteerumite esinemisajad.

Konfigureerimine

- Mõõteintervall.
- Alarmid.

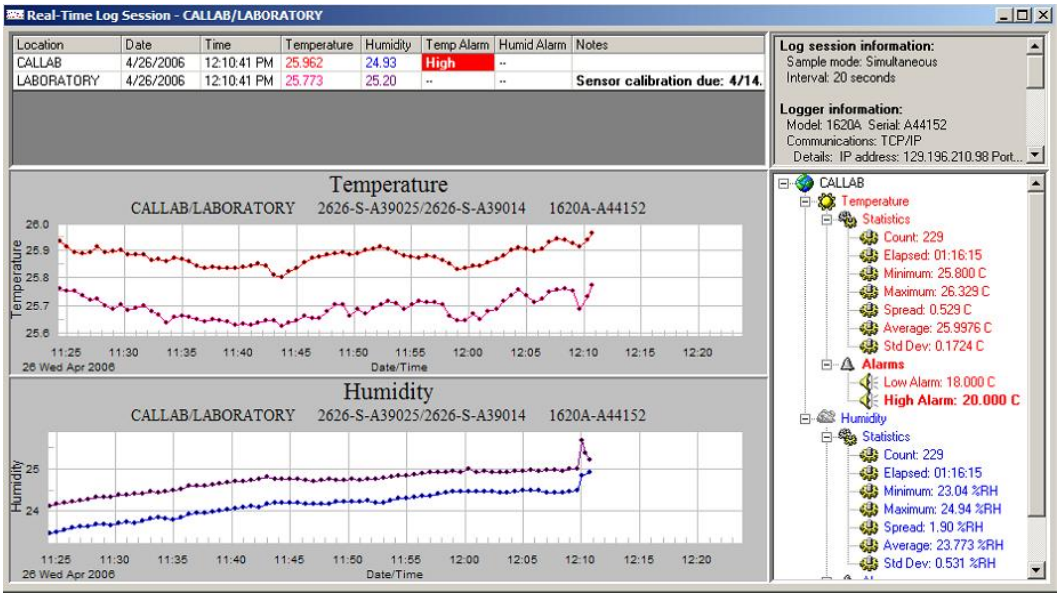
Graafikule kirjete lisamine.

Mitme temperatuurigraafiku korraga näitamine.

3.3. Fluke Calibration - LogWare III

LogWare III [29] on temperatuuri monitoorimise tarkvara.

Järgnevalt joonis LogWare III kasutajaliidesest, võetud LogWare III kasutajajuhendist [30].



Joonis 5. LogWare III reaalaja logi aken.

Võimalused:

- Reaalajas temperatuuri ja/või niiskuse mõõtmine.
- Andmeid saab logida reaalajas üle Etherneti, järjestikliidese või juhtmevaba (WiFi) ühenduse abil.
- Andurit saab „hot-swappida“ – andurit saab vahetada ilma sessiooni lõpetamata.
- Mõõteintervalli valik 1s – 24h.
- Saab määrata mõõtmiste tegemise alg- ja lõppaja.
- Alarmide määramine ja helifaili mängimine kui antud alarm aktiveeritud on.
- E-maili teel erinevate teavituste saatmine.
- Erinevate statistiliste arvutuste tegemine.
- Salvestatud andmete eksportimine erinevatesse failiformaatidesse.

Graafiku kuvamise võimalused:

- Graafikutena saab kuvada erinevaid statistilisi andmeid.
- Graafikute värvi määramine.
- Graafikut saab zoomida.

Raporti kuvamise võimalused, mida raporti peale kuvada saab:

- Alarmide esinemisajad eraldi väljatooduna ja värvi abil teavitamine, kas oli miinimum või maksimum.
- Temperatuuri statistika: mõõtmiste arv, miinimum, maksimum, keskmine ja standardhälve.
- Protsentsuaalselt kui suur osa mõõtmistest oli normi piires ja kui suur osa normist väljas.

3.4. OMEGA® iSD-TC

OMEGA iSD-TC [31] on temperatuuri monitoorimise tarkvara mille GUI on veebipõhine.

Antud juhul on tegu lahendusega, kus temperatuuri logger, mis on omakorda ka server, ühendatakse eemalolevas keskkonnas interneti võrku ja siis saab veebibrauseri abil selle poolt genereeritavaid andmeid vaadata. Seega sarnaneb antud toode veebiruuteriga, mis on mõeldud töötama väga karmides tingimustes ja mille külge saab ühendada täpseid temperatuuri mõõtvaid seadmeid.

Näide tarkvara veebiliideses kuvatavast temperatuurigraafikust [32].

Joonis 6. OMEGA® iSD-TC - Veebiliidese kaudu graafiku kuvamine.

Võimalused:

- Alarmide korral E-maili teel teavituste saatmine.
- Kõige kõrgema ja madalama temperatuuri alarmi sättimine.
- Graafiku koostamine, mida saab veebi teel jälgida.
- Graafiku võimalused:

- X ja Y graafiku dimensioonide valimine.
- Reaalajas olev graafik.
- Graafiku salvestamine ja printimine.
- Mitmete mõõteandmete üheaegne kuvamine.

3.5. Tähtsamad võimalused mida pakutakse kommertsiaalsetes tarkvaralahendustes

Eelpool on antud kerge ülevaade kolmest kommertsiaalsest tarkvaralahendusest, mille abil saab teostada temperatuurimonitooringut.

Nagu näha, siis kõikide lahenduste puhul tähtsamad võimalused olid järgnevad:

- Temperatuurigraafik.
- E-maili teel teavituste saatmine
- Alarmide määramine.
- Mitme temperatuurigraafiku üheaegne kuvamine (nõuab mitut andurit).
- Temperatuuri statistika kuvamine, kas e-maili teel või graafiku peal.
- Kahel juhul suutis tarkvara saata ka e-maili teel raportit.

Kolmas lahendus on aga erinevate võimalustega. Täpsemalt on ta ainult veebist ligipääsetav. Mitmeski mõttes on see hea lahendus nt. serveriruumide monitoorimise puhul ja sellistel juhtudel, kus monitooring toimub kasutajast suurte vahemaade taga.

4. Nõuded arendatavale tarkvaralahendusele

4.1. Programmi nõuete kirjeldamisest

Miks on vaja programmi funktsionaalsusi täpselt kirjeldada.

Tarkvara arendades on mõningad asjad millele peab suurt tähelepanu pöörama, et lõpp-produkt oleks kasutajale mugav, süsteemi mõttes turvaline (ei tekita turvaauke, parimal juhul isegi eemaldab mõne), aga selleks, et antud omadusi implementeerida on vaja kindlat juhist, mis seletaks mida kasutaja/tellija programmilt ootab.

Selleks, et tarkvara luues arenduse protsessi lihtsustada on vaja paika panna nõuded, mida loodav programm peab täitma.

Aga siit tekib küsimus, et kuidas nõudeid kirja panna, sest lihtsalt öelda, et nõue peab seletama mida programm teeb ei ole küll väär, aga see ei ole optimaalne. Kui nõue on mitte-põhjalik, siis võib juhtuda, et arendajad täidavad lüngad omaloominguga, mis võib küll olla just see, mida klient/kasutaja tahtis, aga samas võib see ka kasutajale mitte meelepärane olla. Kõige parem oleks tarkvara arendades selliseid määramatuid olukordi vältida.

Selliste probleemide vältimiseks on otstarbekas kasutada standardeid, mis selgitavad, kuidas tarkvara nõuete kirjeldamine peaks käima.

Üks nendest on IEEE standard „**830-1998 IEEE**“ „Soovitatud praktikad tarkvara nõuete spetsifitseerimiseks“ (ingl. k. *IEEE Recommended Practice for Software Requirements Specifications*) [33], mis on küll asendatud ISO/IEC/IEEE 29148:2011 standardiga. Aga see ei tee temas kirjeldatud soovitatud praktikaid vähemtähtsaks.

Seal on ära toodud, mida üks nõuete spetsifikatsioon peab sisaldama (§4.1). Samuti tuuakse välja hea nõuete spetsifikatsiooni karakteristikud, koos seletustega (§4.3).

IEEE standardis 830-1998 paragrahvis §4.1 on ära toodud soovitused, mida nõuete kirjutaja peab nõuete kirjutamisel addresseerima või arvesse võtma. Antud töö käigus arendatavale tarkvarale on proovitud nõudeid spetsifisteerida just selle paragrahvi abil.

Paragrahvis 4.1 ära toodud soovitused:

- **Funktsionaalsus**
  - Mida tarkvara peab tegema.
- **Välised liidesed**
  - Kuidas tarkvara suhtleb inimestega, riistvaraga, teiste riist- ja tarkvaraliste seadmetega.
- **Jõudlus**
  - Missuguseid nõudeid nõutakse tarkvaralt järgmistelt iseloomujoontelt
    - Kiirus.
    - Kättesaadavus.
    - Reageerimisaeg.
    - Erinevatest funktsioonidest taastumise aeg.
- **Atribuudid**
  - Missugused nõuded on porditavusele.
  - Nõuded täpsusele.
  - Hooldusele.
  - Turvalisusele.
- **Disainist tingitud kitsendused**
  - Kas peab vastama mingitele standarditele?
  - Kas peab olema kirjutatud mingis kindlas keeles.
  - Tingimused andmebaasi terviklikkusele.
  - Piirangud ressurssidele.
  - Piirangud töökeskkonnale.

Järgmises osas pannakse antud soovituste abil paika nõuded antud töö käigus arendatavale temperatuuri monitooringu programmile.

4.2. Nõuded Raspberry Pi-l põhinevale temperatuuri monitooringu programmile

1. Mida tarkvara peab tegema

- Tarkvara peab mõõtma temperatuuri.
  - Seletus - puudub.
- Tarkvara peab mõõdetud temperatuuri salvestama koos ajatempliga logifaili.
  - Seletus – Kasutajal võib tekkida vajadus temperatuuri muutumist hiljem analüüsida.
- Tarkvara peab joonistama temperatuuri muutumise graafiku ajas.
  - Seletus - Parema ülevaate saamiseks temperatuuri muutumisest on visuaalne allikas parem kui tuhat sõna. Näitena võib vaadata antud töö osas 3.3 ära toodud LogWare III graafikut.
- Tarkvara peab olema suuteline arvutama temperatuuri muutumise trendi.
  - Seletus – Parema ülevaate andmiseks temperatuuri muutumisest ajas on vaja lisaks temperatuuri muutumisele teada ka selle trendi.
- Tarkvara peab olema suuteline joonistama trendi graafikut.
  - Seletus – Visuaalne ülevaade on parem kui tuhat koordinaati.
- Tarkvara peab joonistama temperatuurigraafiku ja temperatuuri muutumise trendi graafiku ühele graafikule.
  - Seletus – Parem ülevaade temperatuuri muutumisest ja selle trendist.
- Tarkvara peab omama maksimum ja miinimum temperatuuri alarme.
  - Seletus – Temperatuuri monitooringu tarkvaral (vt. peatükki 3, kus esile toodud näited omavad miinimum ja maksimum temperatuuri alarmide seadistamist) peab olema võimalus alarmide seadmiseks.
  - Kuna kasutajal võib vaja temperatuuri monitooringu programmi kasutada keskkonnas, kus temperatuur peab olema kindlas vahemikus, siis on hädavajalik, et monitooringu programm suudaks eristada töötemperatuuri mitte-töötemperatuurist.
- Alarmide käivitumisel peab tarkvara kasutajale teavituse saatma.
  - Seletus – Kui juhtub, et töökeskkonnas temperatuur väljub normi piirest, siis on vaja sellest kasutajale, näiteks operaatorile, teada anda.
- Tarkvara peab olema suuteline genereerima statistikat kindlast hetkest kindla hetkeni.

    - Parema ülevaate saamiseks on vaja lihtsamat statistikat, et selle abil saaks otsustada, kas temperatuuri muutumist antud ajavahemikus on vaja täpsemalt uurida või mitte.
    - Näite saamiseks vaadata LogWare III ja OMEGA iSD-TC tutvustust.
- Tarkvara peab kasutajat teavitama e-maili teel.
    - Kuna programmi kasutamine peaks olema võimalik ka kõrvalistes paikades (interneti olemasolu peab olema), siis on vaja mingit meetodit, et kasutajat temperatuuri muutustest informeerida.
    - Sama funktsionaalsust omavad eelmises peatükis kirjeldatud kommertsiaalsed tarkvaralahendused.
- Tarkvara peab olema konfigureeritav.
    - Kuna miski siin maailmas ei ole ühesugune, siis antud töökäigus arendatatav programm peab olema ka erinevate keskkondade jaoks kasutajaliidesest konfigureeritav.
- **Konfigureeritavad osad:**
    - Millega temperatuuri mõõta.
        - Kui seadme külge on ühendatud mitu seadet, siis peab saama valida millist välist seadet mõõtmiseks kasutada.
    - **E-maili konfiguratsioon**
        - Kellele saata.
            - Mis kontole e-mail saata („*To:*“ väli e-mailis).
        - Kellelt saata.
            - Kelle kontolt e-mail saadetakse („*From:*“ väli e-mailis).
        - Mis serverit kasutada e-maili saatmiseks.
            - Kuna e-maili saatmiseks on vaja kontot, mis asub mailiserveris, siis peab olema see ka antud tarkvaras sätitav.
        - Mis on saadetava maili pealkirjaks (*Subject*).
            - Kuna temperatuuri monitoorimise seadmeid võib olla mitu, siis oleks vaja eristada erinevatelt seadmetelt tulnud teateid.
                - Üks võimalus eristamiseks on e-maili pealkiri.
            - Kuna võib juhtuda, et mailiserver võib monitooringu maili pealt tulnud kirjad spämmiks arvata, siis peab olema võimalus spämmfiltrile pealkirja põhine filter lisada.

    - Maksimum ja miinimum temperatuuri alarmide suurused.
        - Kuna kasutajaid on mitu ja töökeskkonnad on erinevad, siis võivad erinevad olla ka alarmide suurused, mida tahetakse kasutada.
    - Kui tihti e-maili peale infot saadetakse.
        - Kui kasutaja ei taha, et teda iga tund/päev/nädal tüüdatakse temperatuuri infoga, siis peab olema antud intervall konfigureeritav.
    - Mitu mõõtmiskorda peab temperatuur olema alarmitsoonis, et selle peale kasutajat teavitatakse.
        - Kuna võib juhtuda, et temperatuuriandur võib tõrkuda ja seega võivad mõned mõõtetulemused olla üle ekstreemumite. Seetõttu võib kindluse mõttes olla ekstreemumite piire ületavate mõõtekordade hulk suurem.
- Tarkvara peab salvestama kasutaja poolt määratud konfiguratsiooni.
    - Et tarkvara taaskäivitamisel oleks võimalik laadida kasutaja poolt juba seadistatud konfiguratsiooni, siis peab tarkvara võimaldama selle salvestamist.
- Kui tarkvara uuesti käivitatakse peab tarkvara konfiguratsiooni uuesti laadima.
    - Sellisel juhul ei pea kasutaja iga kord programmi üle konfigureerima.
- **Statistika:**
    - Kindla ajavahemiku:
        - Keskmine temperatuur.
        - Kõrgeim temperatuur.
        - Madalaim temperatuur.
        - Mitu mõõtmiskorda oldi kindlas alarmtsoonis nii madalamas kui ka kõrgemas.
        - Protsentuaalselt kui kaua oldi alarmtsoonis.
    - Näitena võib vaadata LogWare III statistika tööriistariba osas 3.3.
- Kasutajal peab olema võimalik sisestada temperatuuri andurilt saadava info teisendamiseks algoritmi.
    - Seletus: Selleks, et antud tarkvara oleks porditav, st. kasutatav erinevate temperatuurianduritega, peab kasutajal olema võimalus temperatuuri andurilt tuleva info jaoks määrata teisendamise algoritmi, mis siis info teisendaks tavaliseks temperatuuri näiduks.

2. **Välised liidesed**

- **Kuidas tarkvara suhtleb kasutajaga**
  - E-maili teel.
    - Mida e-maili teel saata:
      - Praeguse mõõtevahemiku statistika.
      - Kui tegu on alarmiga, siis öelda, mis alarmiga tegu on.
      - Pilt üldisest graafikust, kus peal on nii trendi graafik kui ka temperatuuri muutumise graafik.
  - Kasutajaliidese kaudu.
- **Kuidas tarkvara suhtleb temperatuurianduritega**
  - Järjestikliidese abil – RS232.

3. **Jõudlus**

- **Piirangud puuduvad**
  - Piirangud küll puuduvad, aga mingit häirivat tarkvara hangumist ei tohiks tekkida.

4. **Atribuudid**

- Peab töötama Linuxi keskkonnas, sest Raspberry Pi-le pakutavad operatsioonisüsteemid põhinevad Linuxi baasil.

5. **Disainist tingitud kitsendused**

- **Piirangud ressurssidele**
  - Peab olema võimeline töötama Raspberry Pi peal.
    - Peab kasutama piisavalt vähe ressursse, et suudaks Raspberry Pi (vähe mälu, tänapäeva mõistes väikese suutliksusega protsessor) peal normaalselt töötada.
    - Kui järgida printsiipi, et arendatav tarkvara võiks töödata ka mobiilsetel seadmetel, siis ollakse arendamisega õigel teel.

4.3. Kasutajaliidese disain

GUI peab olema avar, tavakuva peaks olema täisakna kuva.

Graafik peaks hõlmama suuremat osa tervest programmi vaatest.

Paremal küljel peab nähtaval kohal olema hetketemperatuuri näit. Nupud erinevate seadistuste: e-mail, järjestikliidese ja temperatuuri alarmide konfigureerimiseks peavad olema selgesti arusaadavad.

Avakuval peavad olema ka eraldi nupud mõõtmise alustamise ja lõpetamise jaoks.

5. Võimalikud lahendused Raspberry Pi peal temperatuuri monitooringu tarkvara arendamiseks

Antud peatükis uuritakse ja analüüsitakse, kuidas saaks vastavalt antud töö osas 4.2 kirja pandud nõuetele temperatuuri monitooringu tarkvara Raspberry Pi peal arendada.

Peatüki lõpuks peab selguma, mis raamistikku arendamiseks kasutada, miks seda kasutatakse ja lisaks kuidas Raspberry Pi-d ennast kasutada ehk, mis sisendeid/väljundeid Raspberry Pi-l kasutada ja miks just neid.

Kõigepealt vaatame, mis nõuded on paika pandud – kasutajaliidese, temperatuurigraafiku ja temperatuuri mõõtmise poole pealt.

5.1. Nõuete analüüs

Antud osas analüüsitakse tähtsamaid nõudeid, mis vajavad optimaalseks implementeerimiseks raamistiku tuge.

- Tarkvara peab mõõtma temperatuuri.
    - Raamistik peab pakkuma võimalust USB-ga/järjestikliidesega suhtlemiseks.
- Tarkvara peab mõõdetud temperatuuri salvestama koos ajatempliga logifaili.
    - Meie poolt valitud raamistik peab olema suuteline failidega ümber käia, lisaks peab ta suutma veel süsteemi käest hetkeaega küsida.
- Tarkvara peab reaalajas joonistama aeg/temperatuur graafiku.
    - Raamistik peab omama joonistamise/joonestamise moodulit.
- Tarkvara peab kasutajat e-maili teel teavitama.
    - Raamistik peab omama võrguga suhtlemise võimalust.
- Tarkvara peab olema konfigureeritav.
    - Raamistikul peab olema seadistuste (*settings*) tugi.
- Kasutajaliides
    - Raamistik peab omama võimalust kasutajaliidese arendamiseks.

5.2. Programmeerimiskeeled ja raamistikud

Järgnevalt nimekiri tähtsamatest programmeermiskeeltest ja graafilistest raamistikest, mis töötavad Raspberry Pi-l [34].

Raspberry Pi-l töötavad järgmised programmeerimiskeeled: C#, Java, JavaScript, Pascal, PHP, Python, C/C++.

Tuntumad graafilised raamistikud, mis Raspberry Pi peal töötavad:

- wxPython [35]
  - wxPython on ümbris (*wrapper)* mitmeplatvormilisele graafilisele rakendusliidesele wxWidgets, mis ise on C++ põhine ja kuna ta on mitmeplatvormiline, siis töötab ta erinevate operatsioonisüsteemide peal nt. Windows, Linux/UNIX, Mac OS X.
- QT [36]
  - QT on samuti mitmeplatvormiline graafiline rakendusliides nagu wxPython, aga erineb seetõttu, et QT-s kasutatakse programmeerimiseks C++'i ja QML'i (Javascripti sarnane keel). Aga QT lisab omalt poolt spetsiaalseid makrosid ja spetsiaalse koodigeneraatori, mille abil on QT C++'i rikastanud.

Tegelikkuses on Raspberry Pi-l töötavate graafiliste raamistike arv suurem, aga siin toodi välja need, mida kasutatakse tihedamini ka riistvaralähedases programmeerimises (Python ja C++).

5.3. Graafilise raamistiku valimine

Kuna me arendame graafilist rakendust, siis valiku tegemisel peame lähtuma graafilistest raamistikest.

Valituks osutus QT, sest:

- QT on mitmeplatvormiline
  - Võimeline töötama ka mobiilsetel seadmetel.
- QT on C++ põhine programmeerimiskeel.
- QT käib ajaga kaasas.
  - Hiljuti tulid välja uue ideega: Boot to QT [37].

- QT arendamiseks mõeldud keskkond QT-Creator on mugav ja tarkvaramoodulitega laiendatav [38].
- Kasutajatel on võimalus arenduskeskkonda pluginatega täiendada [39].

Loetelu tähtsamatest QT Creatori laiendustest:

1. Qwt plugin [40], mis lisab QT Creatorile võimaluse kasutajaliidese koostamise osas lisada aknale QWT teegis sisalduvaid graafilisi mooduleid.

2. Doxygen plugin [41], mis genereerib lähtekoodist *doxygen*i kommentaare.

3. Ubuntu QTCreator plugin [42], mis on mõeldud hõlbustamaks Ubuntu arendust QT Creatori peal.

5.4. Sobiva sisendi valimine temperatuuri anduri jaoks

Sisendite tutvustus

Raspberry Pi puhul saab sisendite puhul valida kas USB (vt. USB osa peatükis 2.3) või GPIO (vt. GPIO osa peatükis 2.3) poolt pakutavate võimaluste vahel.

GPIO puhul on mõistlik GPIO-l kasutada järjestikliidese (UART) sisend/väljund kontakte.

USB puhul aga oleks mõistlik kasutada USB – järjestikliides adapterit [43], et siis tarkvaraliselt kasutada USB pesas olevat seadet kui järjestikliidese teel ühendatud seadet.

Järjestikliidese võimalust on mõtekas kasutada sellepärast, et andmevahetust järjestikliidesega on kergem implementeerida võrreldes puhta USB seadmega, lisaks omavad enamused temperatuuriandurid ja mikrokontrollerid just järjestikliidese pesa.

Sisendi valik

GPIO kasutamise teeb ohtlikuks see, et GPIO kontaktid taluvad 3.3V pinget, aga nad ei talu 5V ja kuna Raspberry Pi-l puudub ülepingekaitse, siis on turvalisuse mõttes soovitatud GPIO-d kasutades lisada sellele juurde lisaplaat, millel on olemas puhvrid ja pingetaseme konverteerija, et vältida kiibi läbipõletamist, sest GPIO kontaktid on otseühenduses kiibiga [19].

Seega on turvalisem kasutada sisendiks USB-d tema universaalsuse ja nimetatud turvalisuse pärast.

Antud töös kasutatava temperatuurianduri tutvustus

Antud osas tutvustatakse temperatuuriandurit, mida tarkvara algselt kasutab temperatuuri mõõtmiseks.

Temperatuuriandur põhineb LM50 mikroskeemil [44].

Temperatuurianduri näitajad:

- Mõõtepiirkond: -40 ... 75 ^{o}C
- Mõõtetäpsus: ± 2 ^{o}C

Anduri tööpõhimõte:

Anduri saab arvuti külge ühendada USB juhtme abil. Kui andur on arvutiga ühendatud ja arvutis on tekitatud virtuaalne järjestikport (Raspberry Pi puhul enamasti /dev/ttyACM0), siis hakkab andur temperatuuri väärtuseid üle virtuaalse järjestikliidese edastama ühe sekundilise intervalliga. Antud anduril on ka roheline LED, mis vilgub peale igat temperatuuri väärtuse edastamist.

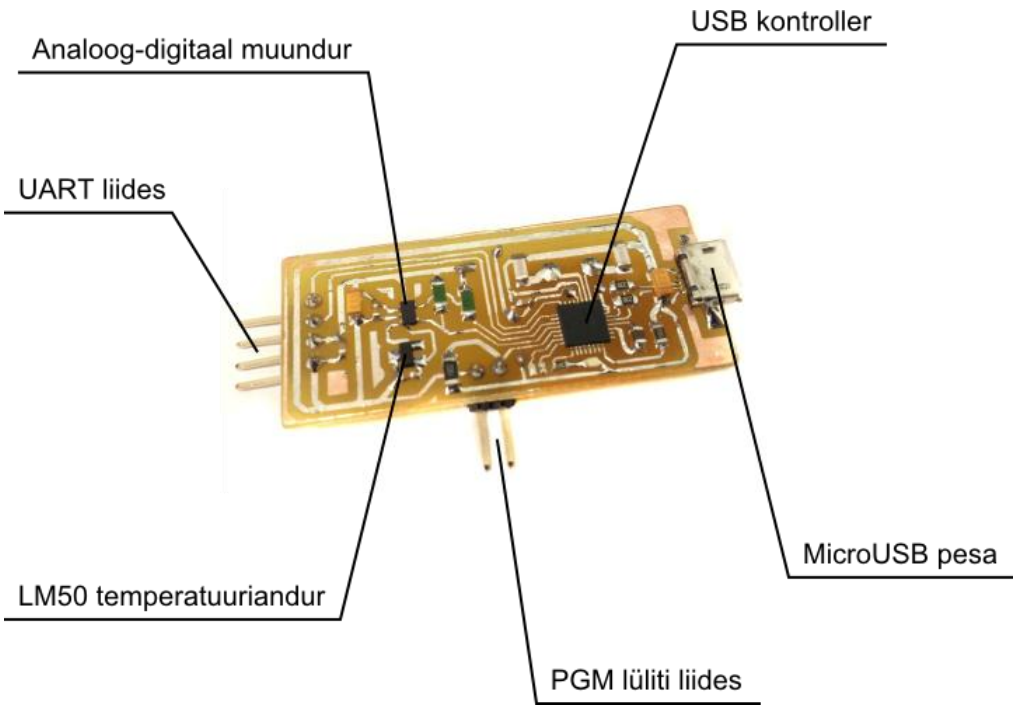
Temperatuuri väärtuse teisendamine

Kuna antud andur saadab väärtuseid otse analoog-digitaal muundurist, siis tuleb need õige temperatuurinäidu kättesaamiseks teisendada.

Andurilt saadud väärtuse teisendamise valem on ära toodud valemis (1).

$$T_{tegelik} = 0.4888 * N_{ADC} - 50 \qquad (1)$$

Järgneval joonisel on välja toodud anduri (valmistatud Aleksei Tepljakovi poolt) struktuur.



Joonis 5. Antud töös kasutatud temperatuuriandur.

5.5. Valikute kokkuvõte

Lõpuks osutus tarkvaraarenduse poole pealt valituks QT raamistik ja sisendite poole pealt USB pesa, kuhu tuleb ühendada USB-järjestikliides adapter, mis virtualiseerib Raspberry Pi-l järjestikliidese ühendust.

6. Tarkvaralahenduse implementatsioon

Antud osas tutvustatakse, kuidas eelmises osas valitud tarkvararaamistikku on kasutatud, et arendatud tarkvaralahendus vastaks antud töös peatükis 4.2 paika pandud nõuetele.

Nagu eelmises osas valitud, siis temperatuuri monitooringu tarkvara arendamise jaoks Raspberry Pi peal kasutatakse QT raamistikku just sellepärast, et ta on uus ja edasiarenev raamistik, mis omab kerget pluginate lisamise võimalust.

6.1. Raspberry Pi operatsioonisüsteem

Antud töökäigus kasutatakse Raspberry Pi peal Raspbian OS-i [45] – Raspbian on Debianil põhinev operatsioonisüsteem, mis on optimiseeritud Raspberry Pi riistvara jaoks.

6.2. QT tutvustus

QT on tarkvaraarendus raamistik, mis baseerub C++ programmeerimis keelel [46].

Tarkvara kirjutamisel toetuti suuresti QT poolt pakutatavatele klassidele.

Antud töös on kasutatud QT versiooni 4.8 [47].

Eelmises peatükis nõuete analüüsis said kirja pandud võimalused, mida raamistik peab pakkuma. Järgnevalt on loetelu QT poolt pakutud klassidest, mis pakuvad nimetatud võimalustest osa:

1. **Meie poolt valitud raamistik peab olema suuteline failidega ümber käia, lisaks peab ta suutma süsteemi käest hetkeaega küsida.**

QT pakub antud probleemi lahenduseks kahte klassi.

QFile [48] – mis on mõeldud failidesse kirjutamiseks ja nendest lugemiseks.

Näide QFile kasutamisest:

Alguses defineerime ära meie poolt kasutatava temperatuuri logifaili.

```
QFile  temperatureLogFile;
```

Seejärel kirjutame sinna faili, kasutades selleks klassi QTextStream [49], mis lihtsustab failidesse kirjutamist ja sealt lugemist:

```
QTextStream out(&temperatureLogFile);
```

Ja seejärel kirjutasime sinna faili.

```
out << "Date:" << tempMeasurement.measurementDate;
```

QDateTime [50] – antud klass võimaldab ajaga ümber käia millisekundi täpsusega. Seda klassi kasutatakse ajatemplite jaoks, lisaks kasutatakse seda ka arve pidamiseks, et teada millal on vaja mingi raport kasutajale saata.

Näide QDateTime kasutamisest:

Kui tahame teada saada praeguse hetke kuupäeva ja aega millisekundi täpsusega, siis saab kasutada järgmist käsku:

```
QDateTime().currentDateTime().toString("dd.MM.yyyy hh.mm.ss.zzz");
```

2. Tarkvara peab olema konfigureeritav.

QT pakub konfiguratsiooni salvestamiseks ühte klassi.

QSettings [51] – antud klass pakub platvormist sõltumatut seadete salvestamist/lugemist.

Näide QSettings kasutamisest:

Kõigepealt luuakse instants QSettings klassist. Kui „settings.ini“ nimeline fail eksisteerib, siis see avatakse, aga kui ei eksisteeri, siis luuakse.

Seejärel loetakse sealt failist sisse suurus, mis asub ’`AlertsDialog::minLevelKey`’ võtme all, see teisendatakse täisarvuks ja salvestatakse muutujasse. Juhul kui sellist võtit konfiguratsiooni failis ei eksisteeri, siis võetakse selle vaikimisiväärtuseks 0.

```
QSettings settings("settings.ini", QSettings::IniFormat);
tempAlertSettings.minLevel = settings.value(AlertsDialog::minLevelKey, 0).toInt();
```

3. **Raamistik peab omama võimalust kasutajaliideste arendamiseks.**

QT pakub kasutajaliidese arendamiseks mitmeid klasse. Kõige tähtsamaks võiks pidada klassi QMainWindow [52], mida kasutatakse peaakna loomiseks, kuhu saab juurde lisada menüüsid, staatuseribasid ja muid QT komponente.

Ülejäänud võimalused

- Raamistik peab pakkuma võimalust USB-ga/järjestikliidesega suhtlemiseks.
- Raamistik peab omama joonistamise/joonestamise moodulit.
- Raamistik peab omama võrguga suhtlemise võimalust.
    - QT küll omab Network klassi, aga antud tarkvara jaoks on siin mõeldud e-maili saatmist.

Nagu eelnevalt näha, siis pole seal neid võimalusi kirjeldatud.

Vastus järgmine – kuna QT olemasolevad klassid/moodulid otseselt selliseid võimalusi ei paku, siis on võimaluseks QT klasse kasutades ise vastavad võimalused programmeerida või kasutada väliseid teeke.

6.3. Koodi taaskasutamise põhjused

Antud osas seletatakse miks antud töös taaskasutatakse teiste poolt loodud tarkvara. Teisisõnu antakse põhjendus väliste teekide kasutamise kohta.

Koodi taaskasutamise kohta (*code reuse*)

Koodi taaskasutamine on tarkvara loomine kasutades juba eksisteerivat tarkvara selle asemel, et seda nullist kirjutada [53].

Miks üldse taaskasutatakse koodi.

Horowitz, E leidis oma töös „An Expansive View of Reusable Software“ [54] juba aastal 1984, et üks fundamentaalsemaid probleeme uute tarkvarade arendamisel on see, et neid hakatakse arendama nullist ja see toob endaga kaasa ajakulu.

Samal ajal oleks aga võimalik kasutada juba varem valmis arendatud tarkvarajuppe ja seeläbi aega ja raha säästa või siis hoopis antud aega kulutada parema ja optimaalsema tarkvara arendamise peale.

Teiseks on suur võimalus, et taaskasutav kood on põhjalikult testitud ja seega saab tarkvaraarendaja veel ajavõitu, sest tarkvara testimine on mahukas protsess.

Lisaks antud töö üks eesmärkidest on näidata, kuidas ja mida saab Raspberry Pi-ga teha ja kui tutvustuse käigus saab näidata, et Raspberry Pi-l saab väga edukalt erinevaid teeke kasutada ja nende abil tarkvaraarendust kiirendada ja midagi mahukamat valmis arendada, siis on see plussiks.

6.4. Väliste teekide kasutamine antud töös

Antud osas tutvustatakse antud töös kasutatud väliseid teeke ja nõudeid, mida antud teekide poolt pakutavad võimalused suudavad täita.

1. **QextSerialPort** [55]

QextSerialPort on QT klasse kasutav teek, mis lihtsustab järjestikliideste kasutamist QT programmides.

Antud projektis kasutatakse seda teeki, et järjestikliidese abil temperatuurianduriga ühendus luua.

Näide teegi kasutamisest

Antud koodijupis kasutatakse QextSerialEnumeratori klassi meetodit getPorts(), et täita järjestikliideste nimekirja süsteemis kättesaadaval olevate portide nimedega.

```
foreach (QextPortInfo info, QextSerialEnumerator::getPorts())
{
  ui->cmbSerialPorts->addItem(info.portName);
}
```

2. **Qwt** [40]

Qwt teek on mõeldud QT-le ja ta omab kasutajaliidese komponente, mis on enamasti kasulikud tehniliste programmide jaoks. Peale 2D graafiku widget-i pakub ta ka muid kasulike lisasid näiteks: skaalasid, liugureid, nuppe ja klahve, kompasse, termomeetreid ja rattaid, mille peal või mille abil saab erinevaid väärtuseid reguleerida.

Antud töö käigus arendatava tarkvara juures kasutatakse Qwt teeki just aeg-temperatuur joonise joonistamiseks ja selle peal temperatuuri muutumise ajas ja temperatuuri muutumise trendi graafikute kuvamiseks.

Näide teegi osade kasutamisest

Kõigepealt määratakse viit QwtPlotCurve-le, mille abil saab graafikule funktsioone joonistada.

```
QwtPlotCurve* tempPlotCurve;
```

Seejärel seotakse antud plot curve vastava joonisega.

```
tempPlotCurve->attach(ui->tempPlot);
```

Seejärel antakse antud graafikule ette tema valim.

```
tempPlotCurve->setSamples(timeData, tempData);
```

Siin on timeData ja tempData kahekordse täpsusega ujukoma arve hoidvad vektorid.

3. **LibQxt** [56]

LibQxt on teek, mis on samuti loodud QT jaoks ja ta lisab funktsionaalsusi QT-le.

LibQxt koosneb järgmistest moodulitest. Loetelu on võetud Qxt dokumentatsioonist [57]:

- **QxtBerkeley –** QT liides Berkely andmebaasi jaoks.
- **QxtCore** – QtCore laiendamise moodul, mis sisaldab mitte-kasutajaliidese funktsioone.
- **QxtGui –** QtGui ja QxtCore laiendused.
- **QxtNetwork –** Laiendab QtNetwork-i ja lisab RPC funktsionaalsust. Samuti sisaldab see moodul e-mailiga suhtlemiseks vajalike klasse.

- **QxtSql –** Moodul, mis laiendab QtSql funktsionaalsust.
- **QxtWeb –** Moodul QT abil võrguteenuste loomiseks.
- **QxtZeroconf** – Antud moodul implementeerib konfiguratsiooni vaba teenuse avastamist.

Antud teegist läheb antud töö käigus vaja ainult QxtNetwork mooduli klasse ja seda just selle mooduli e-mailiga suhtlemist võimaldavate klasside pärast.

Antud moodulist kasutatavad klassid:

QxtSmtp [58]

Antud klass implementeerib SMTP-ga suhtlemist.

Antud töös kasutatakse seda klassi e-maili serveriga suhtlemiseks, sisse logimiseks ja e-mailide saatmiseks.

Näiteid kasutamisest

Kõigepealt luuakse QxtSmtp instants.

Seejärel antakse talle ette mailiserveri kasutajanimi, parool. Seejärel kasutatakse turvalist ühendust, et kasutaja poolt määratud maili serveriga ühendus luua.

```
mailSender = new QxtSmtp(this);
mailSender->setUsername(emailSettings.userName.toAscii());
mailSender->setPassword(emailSettings.password.toAscii());
mailSender->connectToSecureHost(emailSettings.mailServer);
```

QxtMailMessage [59]

Antud klassi kasutatakse e-maili koostamiseks.

Näiteid kasutamisest

Kõigepealt luuakse instants e-mailist.

```
QxtMailMessage     mailMessage;
```

Seejärel täidetakse ära e-maili kohustuslikud väljad: Kellele kiri saata (*To:*), kellelt kiri saadetakse (*From:*) ja e-maili pealkiri (*subject*).

```
mailMessage.addRecipient(emailSettings.sendTo);
mailMessage.setSender(emailSettings.mailFrom);
mailMessage.setSubject(emailSettings.mailSubject);
```

Loomulikult kuulub e-maili juurde ka kirja tekst:

```
mailMessage.setBody(messageBody);
```

Ning saadetakse kiri kasutaja poole teele, kasutades QxtSmtp klassi:

```
mailSender->send(mailMessage);
```

QxtMailAttachment [60]

Antud klassi kasutatakse maili teel manuste saatmiseks. Antud töös arendatavas tarkvaralahenduses kasutatakse seda klassi, et mailile lisada pilt programmi töö käigus genereeritud temperatuurigraafikust.

Näiteid kasutamisest

Kuna antud töös saadetakse kasutajale e-mail koos pildiga, siis on vaja saadetavale e-mailile juurde lisada manus:

```
mailMessage.addAttachment(plotImageFileName,
QxtMailAttachment::fromFile(plotImageFileLocation));
```

4. Octave [61]

GNU Octave on kõrgtasemeline programmeerimiskeel, mis on peamiselt mõeldud numbriliste arvutuste jaoks. Lisaks pakub Octave võimalust graafikaga manipuleerida, näiteks andmeid visualiseerida või pilte manipuleerida.

GNU Octave on nagu vabavaraline MATLAB (ilma kasutajaliideseta) seetõttu on nende peal arendatud programmid kergesti porditavad.

Antud töös kasutatakse Octave-i temperatuuri muutumise trendi arvutamiseks.

Näide Octave-i kasutamisest trendi arvutamiseks

Järgnevas näites koostatakse 2 maatriksit, kus y väärtusteks on temperatuuri suurused ja x väärtusteks on ajahetked, millal temperatuuri suurus mõõdeti.

Kõigepealt koostatakse 2 maatriksit suurustega 1 * ’temperatuuri väärtuste vektori suurus’. Selleks, et programm saaks Octave-i maatriksitele andmeid kergemini ette anda küsitakse maatriksite käest viita nende andmetele. Seejärel kopeeritakse temperatuuri ja ajahetkete andmed Octave-i maatriksitesse.

```
Matrix yTrendData(1, tempData.size());
Matrix xTrendData(1, tempData.size());
double* pyTrendData = yTrendData.fortran_vec();
double* pxTrendData = xTrendData.fortran_vec();
for (octave_idx_type i = 0; i < tempData.size(); ++i)
{
  pyTrendData[i] = tempData.at(i);
  pxTrendData[i] = timeData.at(i);
}
```

Peale seda kui maatriksid on andmetega täidetud, saab alles hakata trendi arvutama.

Selleks kasutatakse Octave-i ’polyfit’ [62] funktsiooni.

Polyfit funktsiooni tutvustus.

p = **polyfit** (*x*, *y*, *n*)

Antud funktsioon tagastab n-astme polünoomi p(x) koefitsentide reavektori, milles on n + 1 muutujat. Antud polünoomi kuju on näidatud valemis (2), kus p_1 on tagastatud vektoris esimene liige ja p_{n+1} viimane liige.

$$p(x) = p_1x^n + p_2x^{n-1} + .. + p_nx + p_{n+1} \qquad (2)$$

Antud polünoom on leitud sobituma etteantud (x, y) punktidega kasutades vähimruudu meetodit.

Vähimruudu meetodil leitud polünoom vastab oma omadustelt valemile 2 (polünoomi kordajad leitakse selliselt, et nad minimiseeriksid viga $p(x_i)$ ja y_i vahel). Vähimruudu meetod on ära toodud valemis (3) [63].

$$\sum_{i=1}^{n}[y_i - f(x_i)]^2 = min \quad (3)$$

Järgmises koodijupis tehakse octave_value_listist instants ja täidetakse see ära suurustega, mida on vaja, et saaks polyfit-i kasutada.

Antud töös kasutatakse *polyfit*-i lineaarse polünoomi leidmiseks, mis on ära toodud valemis (4).

$$y = ax + b \quad (4)$$

Kui kasutada Octave-i teeke C++ programmis, siis „*polyfit*“ funktsiooni teek ei sisalda, vaid selle funktsiooni kasutamiseks tuleb kasutada „*feval*“ nimelist [64] funktsiooni.

```
octave_value_list inputValues;
inputValues(0) = xTrendData;
inputValues(1) = yTrendData;
inputValues(2) = 1;
octave_value_list result = feval("polyfit", inputValues);
```

„*polyfit*“ tagastab meile reavektori vastavate polünoomi väärtustega, kus esimene väärtus on polünoomi n-nda astme kordaja ja viimane väärtus on 0-nda astme kordaja.

Järgnevalt väike koodijupp nende väärtuste kättesaamisest.

Kõigepealt kontrollitakse, kas „*feval*“ funktsioon vastus sisaldab tulemust.

Kui jah, siis tekitatakse tulemuse listist maatriks, seejärel kontrollitakse, kas maatriksis on rohkem kui 1 element. Kui see on ka tõene, siis salvestatakse tulemuse väärtused vastavatesse muutujatesse nende hilisemaks kasutamiseks.

```
if (false == result.empty())
{
    Matrix output(result(0).matrix_value());
    if (output.numel() > 1)
    {
     trendLineFirstParam = output.elem(0);
     trendLineSecondParam = output.elem(1);
```

6.5. Kokkuvõte teekide kasutamisest antud töös

Nagu näha, siis kasutades väliseid teeke said kõik tähtsamad nõuded, mida QT raamistik ei võimaldanud lihtsamal viisil lahendada, lahendatud.

Kui QextSerialPort, Qwt ja LibQxt olid kõik mõeldud QT raamistikule lisafunktsionaalsuste andmiseks, siis Octave on puhtalt väline teek.

Võib küll arvata, et milleks ühe funktsiooni, antud juhul „*polyfit*“, kasutamiseks on vaja tervet Octave-i paketti kasutada, kui võib olla oleks lihtsam trendi üheastmeline polünoom ise algoritmi kasutades välja arvutada, aga see pole töö peamine mõte. Antud töö üks eesmärkidest on näidata, et kuidas ja mis tarkvara saab Raspberry Pi peal arendada.

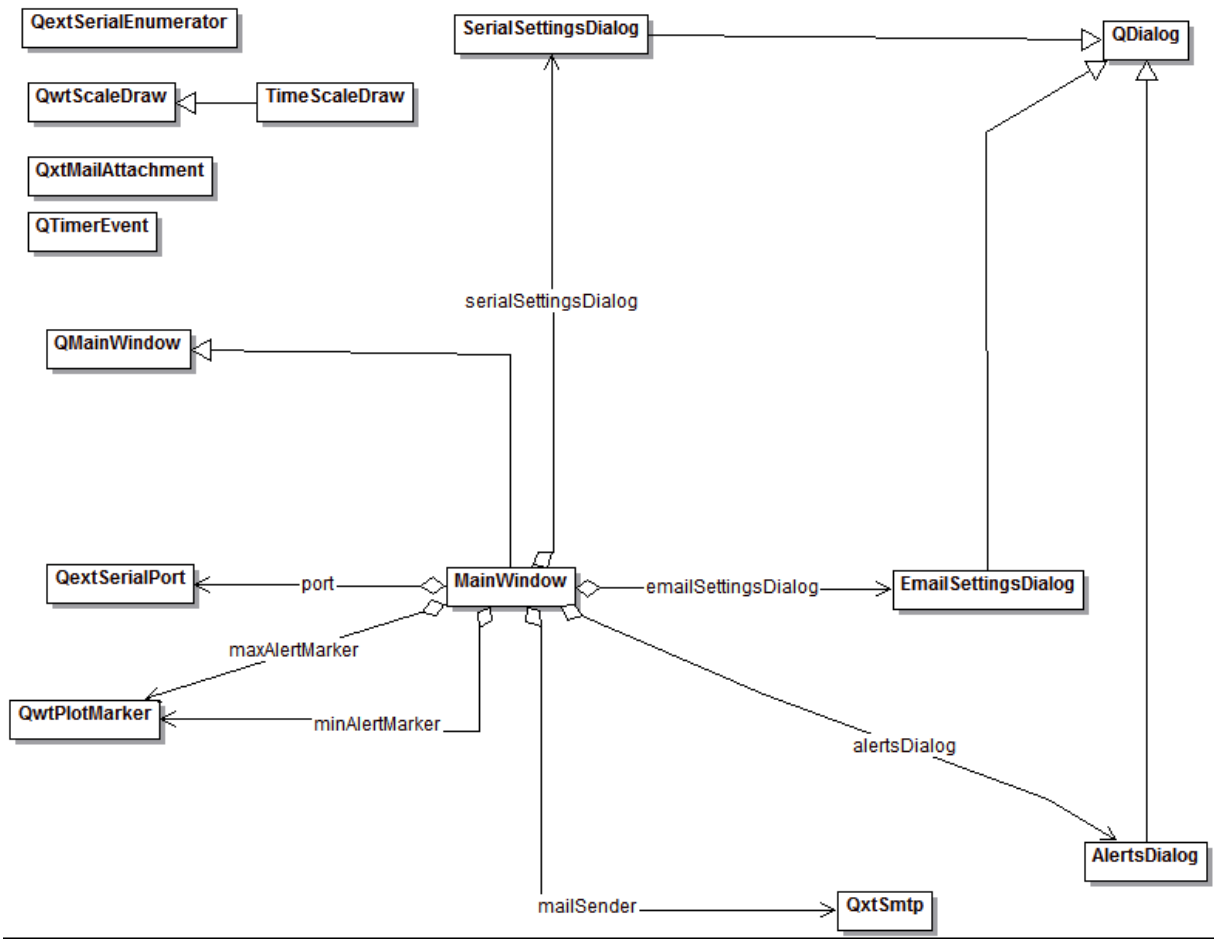
7. Tarkvaralahenduse struktuur

Antud peatükis tutvustatakse lähemalt antud töö tarkvaralahenduse klasse ja kuidas neid osi kokku pannes tekkis rakendus.

7.1. Klasside tutvustus

Klasside hiearrhia

Kõigepealt on näidatud ära erinevate dialoogide pärinemine QDialog klassist ja üldine klasside kasutamine antud tarkvaralahenduses.

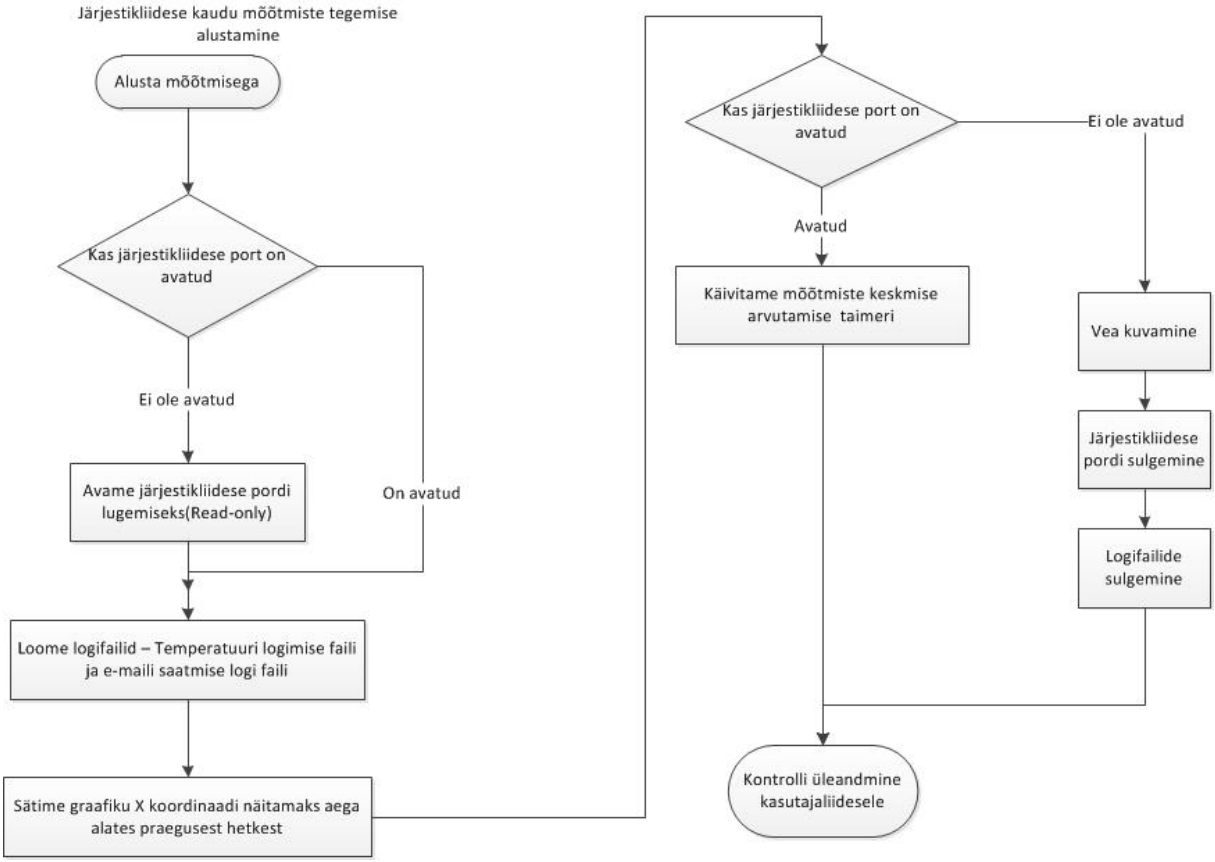


Joonis 6. Klassidiagramm töös kasutatud klassidest.

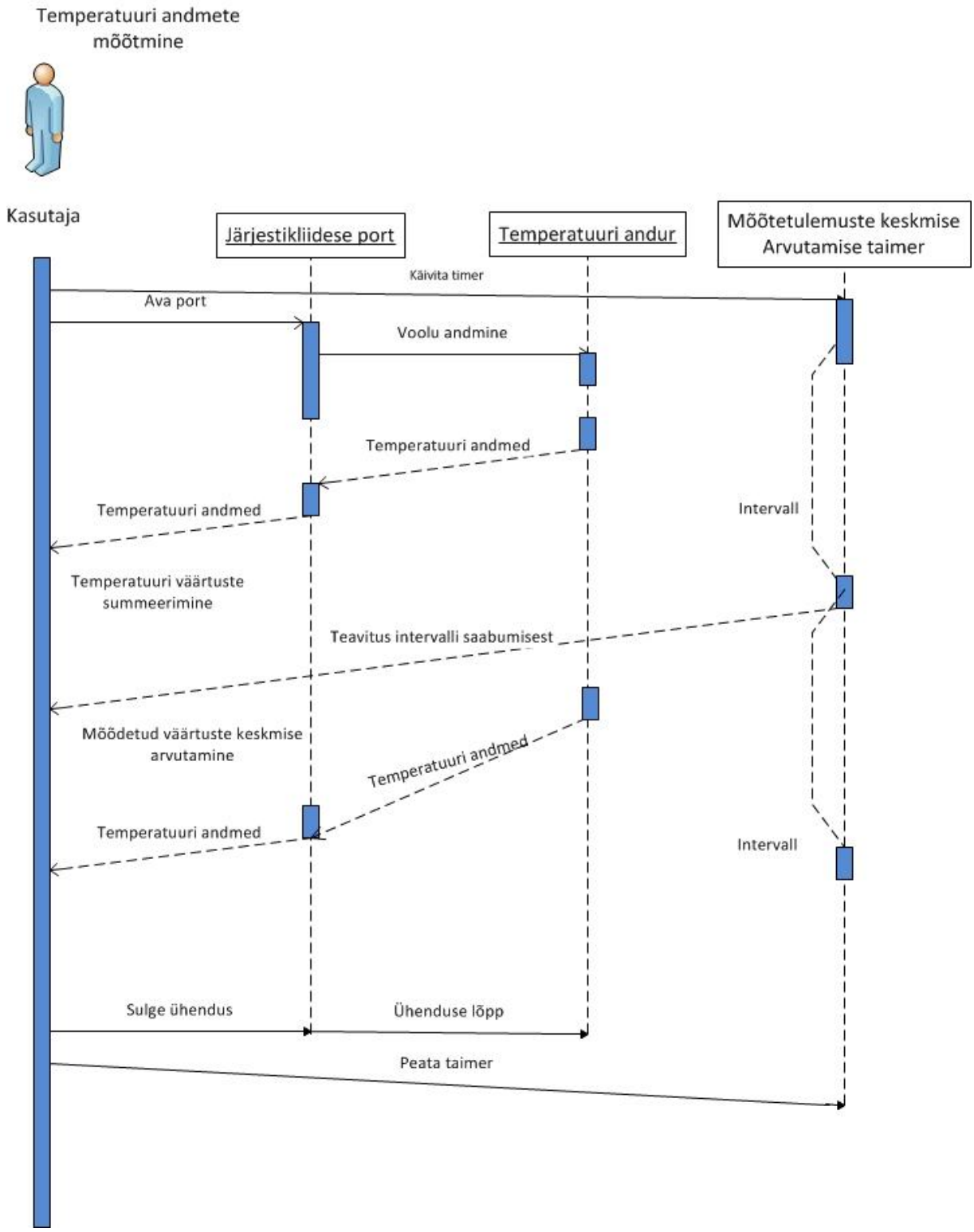
7.2. Programmi töö diagrammid

Järgnevalt skeemid programmi töö kohta.

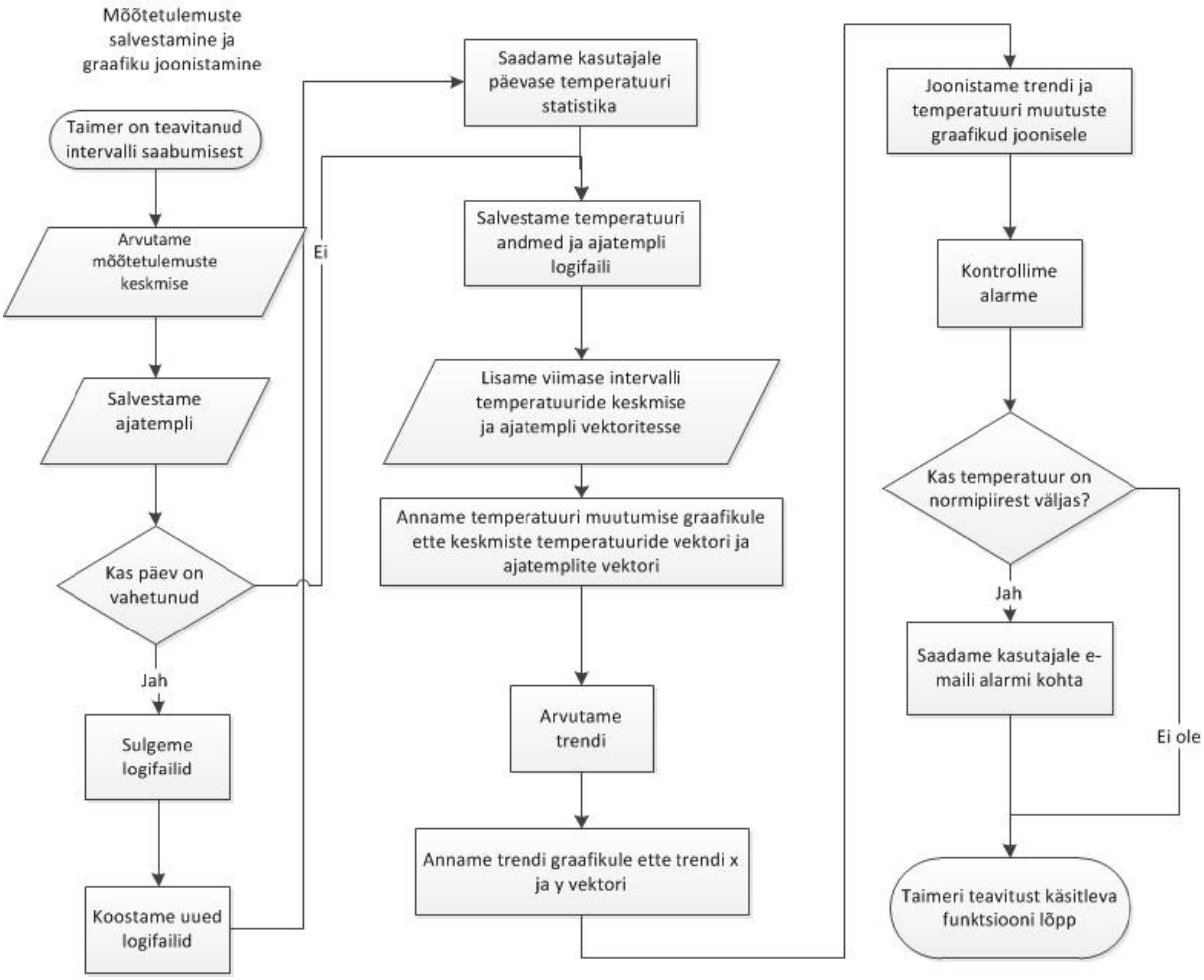
Joonis 7. Voodiagramm: Programmi käivitamine.



Joonis 8. Voodiagramm: Mõõtmiste tegemise käivitamine.



Joonis 9. Jadaskeem: Andurilt temperatuuri andmete saamine ja taimeri intervallid.



Joonis 10. Voodiagramm: Mõõtetulemuste salvestamine ja graafikute joonistamine.

7.3. Programmi lähtekoodi failide tutvustus

- **main.cpp** – Programmi käivitamine. Antud moodulis käivitatakse UI ja Octave initsialiseeritakse.
- **mainwindow.cpp/.hpp** – Peaakna moodul – klassidiagrammil MainWindow.
  - Peaakna moodulis toimub kõik peaaknaga seotud:
    - Joonise haldamine.
    - E-Mailide saatmise haldamine.
    - Temperatuuri väärtuste vastuvõtmine andurilt ja nende salvestamine ja analüüsimine.
    - Algne programmi konfiguratsiooni laadimine seadete failist.
- **alertsdialog.cpp/.hpp** – Alarmi konfiguratsiooni akna moodul – klassidiagrammil AlertsDialog.
  - Antud moodulis toimub:

        - Alarmi seadete muutmine/haldamine.
        - Alarmide salvestamine konfiguratsiooni faili.
- **emailsettingsdialog.cpp/.hpp** – E-Mailiga seotud seadistuste akna moodul – klassidiagrammil EmailSettingsDialog.
    - Antud moodulis toimub:
        - Emaili seadete muutmine/haldamine.
        - E-maili seadete salvestamine konfiguratsiooni faili.
- **serialsettingsdialog.cpp/.hpp** – Järjestikliidesega seotud seadistuste akna moodul – klassidiagrammil SerialSettingsDialog.
    - Antud moodulis toimub:
        - Saadaval olevate järjestikliideste portide kuvamine kasutajale.
        - Ühendatava järjestikliidese pordi seadete muutmine/haldamine.
        - Järjestikliidese seadmete muutmine/haldamine.
- **ui_* algusega päisefailid** – Antud failid on kasutajaliidese failid, mis on kasutajaliidest tehes QT Creator-i poolt genereeritud.

8. Tarkvaralahenduse tutvustus

Antud peatükis kirjeldatakse, kuidas arendatud tarkvara saab Raspberry Pi peale kopeerida ja mis teeke on vaja alla laadida, et programmi saaks Raspberry Pi peal kompileerida.

Seejärel tutvustatakse programmi väljanägemist ja näidet selle kasutamisest.

8.1. Programmi transportimine Raspberry Pi peale ja selle kompileerimine

Enne kui kompileerima hakata tuleb seletada kuidas antud tarkvaralahenduse lähtekoodi Raspberry Pi peal saab kompileerida.

Kõigepealt tuleb Raspberry Pi-le installida paar teeki. Selleks tuleb Raspbiani terminalis käivitada järgmised käsud:

Järgmise käsuga laetakse tsentraalsest serverist uuendused programmidele mida on apt-get-ga installitud.

```
sudo apt-get update
```

Seejärel tuleb installitud programmid uuendada.

```
sudo apt-get upgrade
```

Seejärel tuleb installida järgmised tarkvarapakid:

- qt-sdk – QT4 arendamise jaoks mõeldud programmid, teegid ja definitsioonid.
- libqwt-dev – QWT arendamise jaoks mõeldud teegid ja definitsioonid.
- liboctave-dev – Octave arendamise jaoks mõeldud teegid ja definitsioonid.
- libqxt-dev – Qxt arendamise jaoks mõeldud teegid ja definitsioonid.
- libssl-dev – Kuna QxtSmtp võimaldab luua ka turvalist ühendust maili serveriga (GMail toetab ainult turvalist ühendust), siis on vaja installida ka see pakk.

```
sudo apt-get install qt-sdk libqwt-dev liboctave-dev libqxt-dev
libssl-dev
```

Nüüd kus pakid on installeeritud saab alustada lähtekoodi Rasperry Pi peale liigutamisega.

Kuna ise kasutasin programmi kirjutamiseks Windowsi keskkonda, siis leidsin, et kõige parem on failide ülekandmiseks WinSCP, mis on avatud lähtekoodiga tasuta SFTP, SCP, FTPS, FTP klient Windowsi jaoks [65].

Sellega lõin SFTP ühenduse kohalikku võrku ühendatud Raspberry Pi ja minu Windowsi keskkonna vahel.

Kui failid ülekantud, siis saab hakata kompileerima.

Kõigepealt tuleb lasta qmake-l genereerida Makefile, selleks tuleb lähtekoodi kaustas, kus asub fail nimega TemperatureWatcher.pro, käivitada järgmine käsk:

```
qmake TemperatureWatcher.pro
```

Seejärel tuleb programmi kompileerimiseks ja linkimiseks kasutada järgmist käsku:

```
make
```

Nüüd kui peale make-i kasutamist on programm kompileeritud saab ta käivitada:

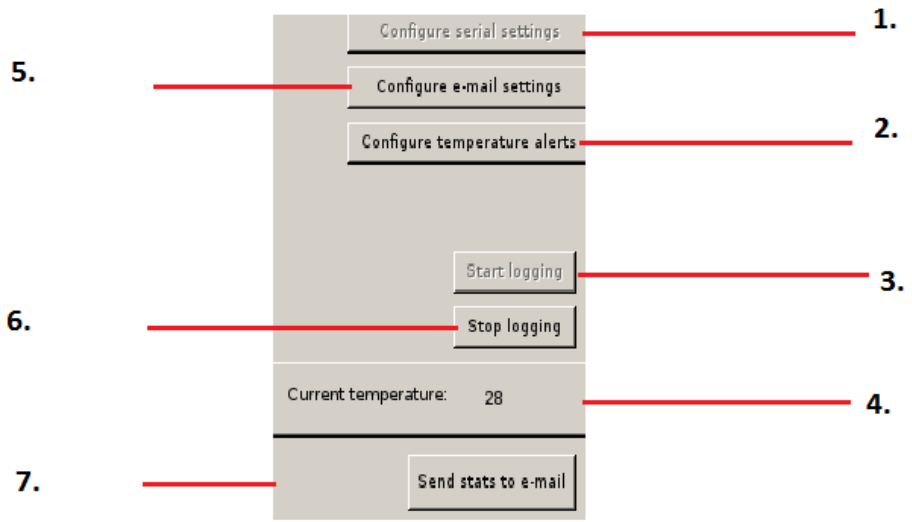
```
./TemperatureWatcher
```

Järgmises osas täpsemalt, mis juhtub kui eelmine käsk käivitada.

8.2. Kasutajaliidese peaaken

Antud osas tutvustatakse tarkvaralahenduse peaakent.

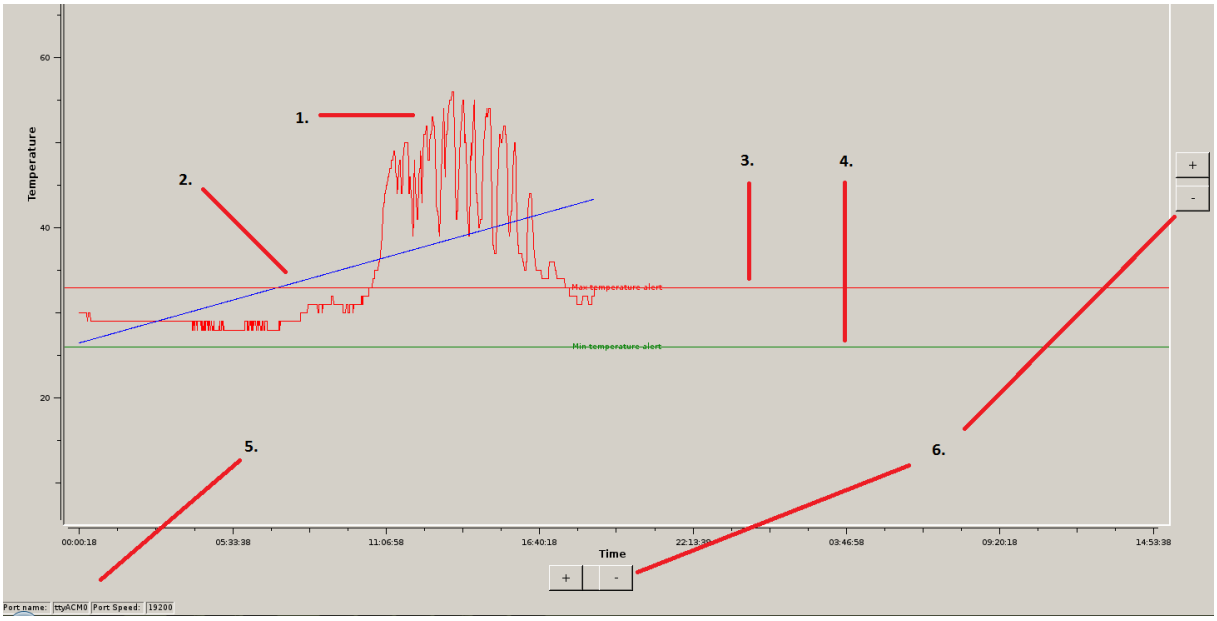
Kõigepealt kontrollpaneel:



Joonis 11. Temperatuuri monitooringutarkvara kontrollpaneel.

- **1. Järjestikliidese seadistuste dialoogi nupp.**
    - Kui programm on töötamas, siis järjestikliidese seadeid ei saa muuta.
- **2. Temperatuuri alarmi seadistuste dialoogi nupp.**
- **3. Logimise alustamise nupp.**
    - Kui kasutaja on juba logimisega alustanud, siis seda nuppu mõõtmiste tegemiste ajal vajutada ei saa.
- **4. Hetke temperatuuri näit.**
    - Hetke temperatuuri näidatakse alati kui andurilt tuleb uus temperatuuri väärtus.
- **5. E-Maili seadistuste dialoogi nupp.**
- **6. Logimise peatamise nupp.**
    - Kui logimine peatatakse, siis suletakse ka logifailid ja saab jälle vajutada nuppu nr. 1 ja nr. 3.
- **7. Statistika e-maili peale saatmise nupp.**
    - Kui antud nuppu programmi töökäigus vajutada, siis tehakse hetke temperatuuri muutumise graafikust pilt ja saadetakse see koos hetkestatistikaga kasutaja poolt etteantud e-mailile.

Peaakna temperatuuri ajas muutumise graafik (PS! Kõrged temperatuurid on tingitud anduri asetamisest aknalauale):

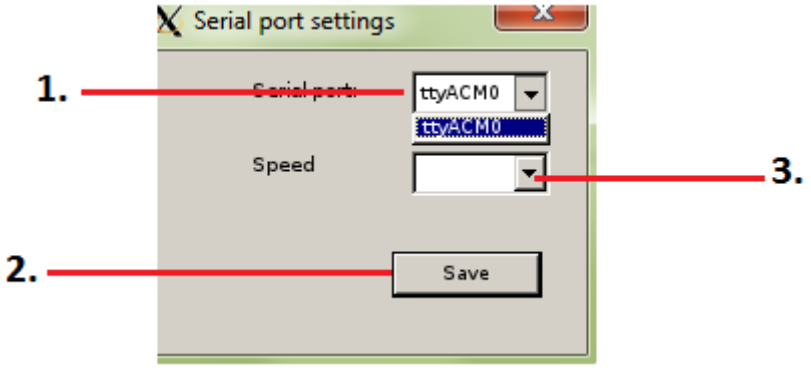


Joonis 12. Peaakna temperatuurigraafik.

- **1. Punane joon – Temperatuuri joon.**
- **2. Sinine joon – Temperatuuri trendi joon.**
- **3. Maksimum temperatuuri alarmi joon.**
- **4. Miinimum temperatuuri alarmi joon.**
- **5. Teave hetkel valitud järjestikliidese pordi kohta: Pordi nimi ja kiirus.**
- **6. X ja Y graafikute sisse- ja välja zoomimise nupud.**

8.3. Konfiguratsiooni dialoogide tutvustus

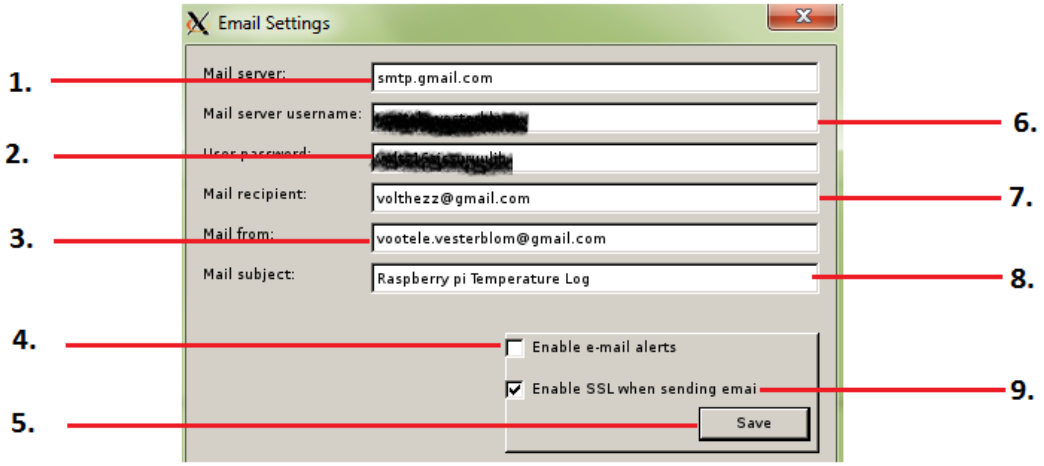
Järjestikliidese seadete dialoog:



Joonis 13. Järjestikliidese seadete dialoog.

- **1. Valitav järjestikliidese port millega ühendus luua, et sealt temperatuuri andmeid saada.**
- **2. 'Save' nupp.**
  - Selleks, et seadistused salvestuksid ja kajastuksid inforibal (Joonis 12. Peaakna temperatuurigraafik. p. 5) tuleb seda nuppu vajutada.
- **3. Järjestikliidese ühenduse kiiruse valimine.**
  - Praegu saab valida kolme kiiruse vahel: 4800, 9600 ja 19200 baudi.

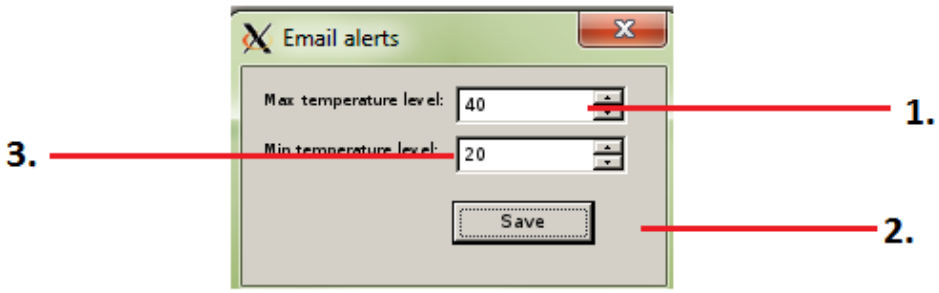
E-Maili seadete dialoog:



Joonis 14. E-Maili seadete dialoog.

- **1. E-Mailiserver.**
  - Mailiserver, kus asub kasutaja e-maili konto, kuhu programm ühendub, et e-maili saata.
- **2. Kasutaja parool.**
  - Kasutaja parool antud mailiserveris.
- **3. „From:“ väli tavalistes e-maili rakendustes.**
  - Maili saaja näeb, et sellelt kontolt on mail tulnud. Nt. kasutaja saab selliselt saatjalt tulnud e-maile filtreerida, et neid spämmiks ei loetaks ja/või neid eraldi kausta tõsta.
- **4. Linnuke siia, kui tahetakse, et e-maili peale saadetaks alarmi korral teavitus.**
- **5. Salvesta seadistus. Kui ’Save’ nuppu ei vajutata, siis sisestatud sätetega maili ei saadeta.**
- **6. Kasutajanimi.**
  - Kasutajanimi mailiserveris, kuhu programm ühendub, et maili saata.
- **7. Kellele kiri saata.**
- **8. E-Maili pealkiri.**
- **9. Linnuke siia kui tahate e-maili ühenduse ajal turvalist ühendust.**
  - Gmaili kontode korral peab siin linnuke olema, muidu tuleb error ja kirja ei saadeta.

Järgmisena alarmide seadistamise dialoog:



Joonis 15. Alarmide dialoog.

- **1. Maksimaalse alarmi seadistamine.**
- **2. Salvestamine.**
  - Kui 'Save' nupp jääb vajutamata, siis just sätitud alarme ei salvestata.
  - 'Save' nuppu vajutades tekivad graafikule vastavalt alarmide väärtustele neid tähistavad jooned.
- **3. Minimaalse alarmi seadistamine.**

8.4. Näide programmi poolt saadetavast e-mailist

Antud osas on toodud näide päevase statistika e-mailist, mis kasutajale saadetakse päeva vahetumisel.

„

Info from temperature sensor:

Sending Daily Temperature statistics.

Daily statistics at the moment are:

Daily minimum is: 28

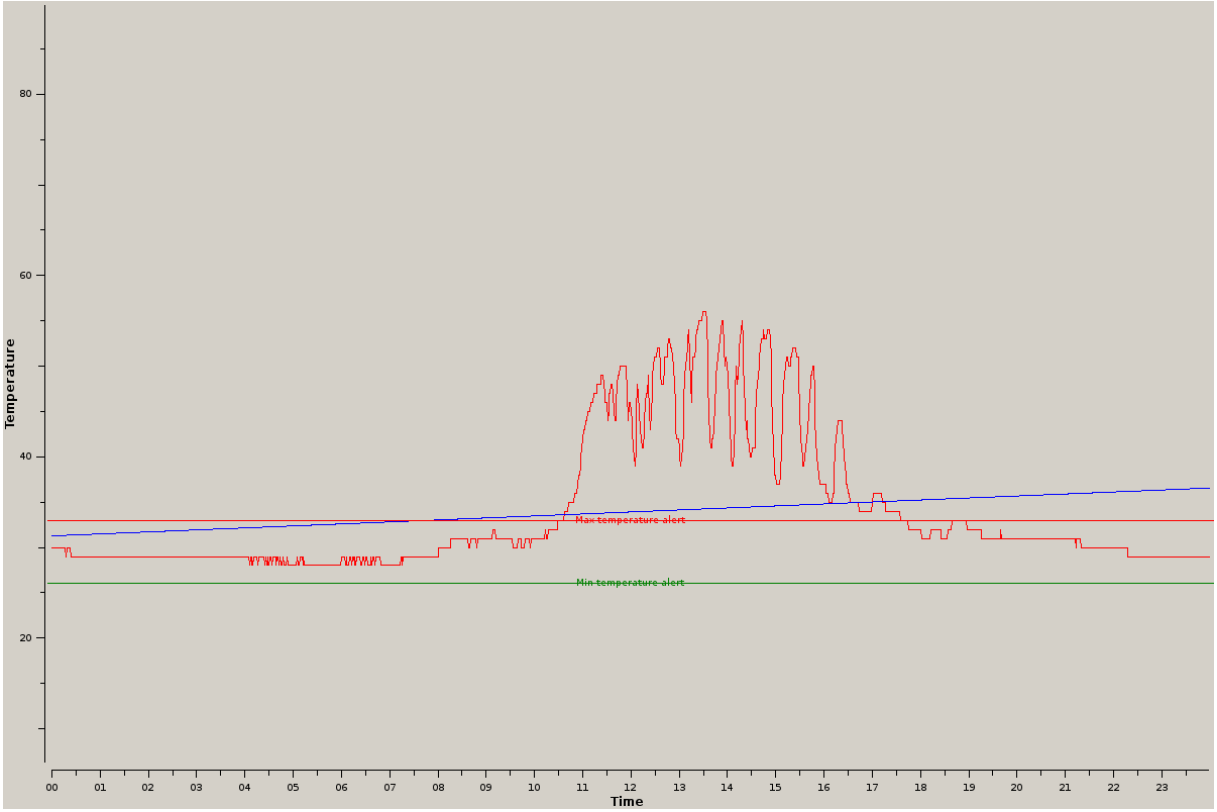
Daily maximum is: 30

Daily average is: 29

*Trendline formula is: -6.67222e-06*x + 29.3936*

„

Millele on manusena kaasa pandud näiteks järgmine joonis, mis kujutab endas temperatuuri muutumist päikselisel päeval:



Joonis 16. E-Mailile saadetud päevase statistika graafik.

9. Edasiarendamise võimalused

Antud osas tutvustatakse ideid, mis jäid antud töös realiseerimata, aga võiksid olla kunagi antud tarkvaralahenduses implementeeritud.

9.1. HeaterLabKitiga integreerimine

Idee on HeaterLabKiti [66] kontrollimiseks arendada antud töö käigus arendatavale tarkvarale lisamoodul, mis kuvaks HeaterLabKitilt saadud temperatuuri näitu ja samas omaks võimalust KITi kontrollida.

Kontrollida võiks seda PID-kontroller. Selleks peaks antud idee lisamoodulil olema implementeeritud ka PID-kontroller, mis oleks siis kasutaja poolt reguleeritav.

Peamine mõte antud idee juures on see, et kuna HeaterLabKit on mõeldud õppematerjalina kasutamiseks, siis saaks õpilastele samal ajal ka Raspberry Pi-d õpetada ja mõlema koostöös saaks õpilastele PID-kontrollereid õpetada (mida erinevad osad PID kontrolleris teevad).

Eeldatav lahendus

Antud probleemi on võimalik praegu olemasolevate teekida lahendada.

HeaterLabKiti temperatuuri joonise kuvamiseks saab kasutada sama moodulit, mida on kasutatud antud töös temperatuuri muutumise graafiku kuvamiseks.

PID juhtimist saab implementeerida Octave-i abiga. On vaja kirjutada .m funktsioon, mida siis feval funktsioonile ette anda vastavate parameetritega.

Ning selleks et HeaterLabKiti kontrollida peame HeaterLabKitiga suhtleva pordi avama Loe/Kirjuta seadistuses.

9.2. Lisa graafikute lisamine .m failide abil

Idee

Lisamoodul antud tarkvaralahendusele, mis võimaldab täiendavaid temperatuuri analüüse joonisel kuvada.

Kuidas

Anname kasutajale võimaluse lisada kasutaja poolt valmistatud .m faile, mille väljundväärtuste abil genereeritud graafikutele saab kasutaja määrata värvi, nime, joone paksuse ja joone kuju/vormi.

Lisaks kirjeldame ära mida .m fail peab sisendiks võtma ja mida väljundiks.

Takistused

Praegusel juhul tuleb tarkvara lähtekoodi refaktoreerida, et oleks hõlpsam antud ideed moodulina arendada.

Kuidas arendada

.m failide käivitamine on puhtalt Octave-i feval funktsiooni pärusmaa ja kui me anname ette kriteeriumid, et millised peavad olema .m faili sisendid ja väljundid (kas maatriks või vektor või arv vms.) siis on see probleem lahendatav.

Antud hetkel kasutuses olevaid teeke kasutades on võimalik antud ideed realiseerida.

9.3. Anduri temperatuuri võrdlemine meteoroloogiliste andmetega

Idee

Laseme programmil programmis ettemääratud meteoroloogilist infot pakuvatelt veebisaitidelt temperatuuri infot koguda, seejärel info logisse salvestada kõrvuti koos meie temperatuuri mõõtmise logiga.

Seejärel saaks kasutaja näiteks temperatuuri logi ja meteoroloogiliste andmete logi kasutades erinevaid analüüse jooksutada.

Takistused

Pole nimekirja meteoroloogilistest serveritest, kes lihtsal kujul annavad temperatuuri infot välja koos ajatemplitega.

Kuidas arendada

Antud juhul mängiks rolli QxtNetwork moodul.

9.4. Veebiliidese kaudu anduri(te)lt info kätte saamine

Idee

Idee on analoogne näidete osas ära toodud OMEGA iSD-TC-ga (vt. osa 3.4).

Idee on järgmine:

Kasutajal on mitu Raspberry Pi-ga temperatuuri mõõtvat seadet, mis on kõik internetti ühendatud. Kõik seadmed on temperatuuriserverid, mis võrku ühendades annavad oma võrku ühendumisest näiteks tsentraalsele serverile, mis on juba ette konfigureeritud, teada.

Kommunikatsiooni alustamine tsentraalserveriga võib toimuda mitut erinevat moodi. Üks võimalus oleks näiteks e-maili teel tsentraalserverile teada anda oma IP aadress ja ID, mille külge tsentraalserver saab ühenduda, et mõõteseadet tsentraalserveri IP-st teavitada. Antud meetod on kasulik sellistel juhtudel, kus tsentraalserveri IP võib muutuda.

Lisaks võiks tsentraalserver pakkuda veebipõhist kasutajaliidest, mille kaudu saab erinevate mõõteseadmetelt tulnud infot lugeda ja analüüsida.

Lisaks võib kasutajal olla võimalus veebiliidese kaudu kontrollida serveri külge ühendunud mõõteseadmeid. Näiteks: mõõtmise alustamine/lõpetamine, intervallide määramine (info edastamise intervall, mõõteintervall jne.), teave uuest serveri e-mailist millele tuleb endast teada anda.

Takistused

Tuleb välja uurida, kuidas andmeid serverisse saadetakse; kuidas veebiliidese kaudu mõõteseadet kontrollida; turvalisus.

Kõik see nõuab täiendavaid ressursse.

Kuidas

Antud idees on kaks osa mida on vaja lahendada:

- Raspberry Pi tarkvaralahenduse edasiarendamine.
- Veebiliidese arendamine.

Raspberry Pi poolset tarkvaralahendust saab olemasolevaid teeke kasutades edasiarendada (Qxt ja selle moodul QxtNetwork), aga veebiliidese arendamine nõuab teist laadi oskusi, mida QT raamistiku kasutades optimaalselt lahendada ei saa, vaid vajab veebiprogrammeerimise oskust.

10. Kokkuvõte

Hirmu, et noortele arvutiteadus, programmeerimine ja muud tehnoloogilise loomega tegelevad alad tulevikus enam ei meeldi võis hiljaaegu täitsa normaalseks ja objektiivseks pidada. Nüüd on aga status quo muutumas, sest väike, aga suutlik Raspberry Pi on turule tulnud ja tema põhieesmärgiks on tuua noored tagasi infotehnoloogilise loometegevuse juurde. Olgu selleks loomeks robotid, väikesed LED-e vilgutavad programmid või siis hoopis mõned kompleksed süsteemid, siis võib ainult loota, et tulevikus toob Raspberry Pi infotehnoloogilise loome juurde aina rohkem ja rohkem tegusaid noori.

Antud töö peamiseks eesmärgiks oli temperatuuri monitooringu süsteemi arendamine Raspberry Pi peal. Selle käigus tutvustati ka Raspberry Pi võimalusi tarkvaraarenduse poole pealt.

Tarkvaraarenduse käigus kasutatud vahendite abil on aga võimalik juba valminud rakendust täiendada, seda just hästi valitud tarkvaraarenduse raamistiku (QT) pärast, mis on arenemas väga mitme-platvormiliseks tarkvaraarenduse keskkonnaks. Kui veel lisaks vaadata, milliseid väliseid teeke kasutati antud töös arendatava tarkvaralahenduse implementeerimisel, täpsemalt GNU Octave-i kasutamist, siis võib öelda, et Raspberry Pi arenduse võimalused ainult selle näitega ei piirdu, sest Octave pakub väga suurt võimalust Raspberry Pi peal midagi suurejoonelist/huvitavat/komplekset arendada.

Kokkuvõttes võib öelda, et vaadates kuidas Raspberry Pi müüginumbrid on jõudsalt peale müüki tulemist tõusnud ja kuidas „Raspberry Pi“ Google otsingus trendib jääb üle ainult huviga oodata ja vaadata, kuidas ta hakkab noori innustama ja mida Raspberry Pi kasutajad selle peal arendama hakkavad, sest arendusvõimalusi millegi innovaatilise loomiseks on tal piiritult.

Kasutatud kirjandus

[1] Kolmveerandil leibkondadest on kodus internetiühendus. Eesti Statistika. [WWW] http://www.stat.ee/57583 (28.05.13)

[2] Jaagant, U. IT-valdkond võitleb tööjõupuuduse vastu kõikide võimalike vahenditega. – Eesti Päevaleht, 2013, 9. mai. [E-ajaleht] http://www.epl.ee/news/eesti/it-valdkond-voitleb-toojoupuuduse-vastu-koikide-voimalike-vahenditega.d?id=66097536 (28.05.13)

[3] About us. - Raspberry Pi [WWW] http://www.raspberrypi.org/about (18.05.13)

[4] Computer Laboratory. Raspberry Pi. – Universitiy of Cambridge [WWW] http://www.cl.cam.ac.uk/projects/raspberrypi/ (18.05.13)

[5] BBC Microcomputer System User Guide [WWW] http://regregex.bbcmicro.net/BPlusUserGuide-1.07.pdf (26.05.13)

[6] BBC Co. UK. – „The BBC Microcomputer and me, 30 years down the line“ [WWW] http://www.bbc.co.uk/news/technology-15969065 (18.05.13)

[7] Mitchell, G. „The Raspberry Pi single-board computer will revolutionise computer science teaching [For & Against]“, *Engineering & Technology*. 2012, 7 (3), 26. [*Online*] http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6210296&isnumber=6198714 (26.05.13)

[8] Bizony, Piers. "The Raspberry Pi single-board computer won't revolutionise computer science teaching [For & Against]," *Engineering & Technology*. 2012, 7 (3), 27 [*Online*] http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6210297&isnumber=6198714 (26.05.13)

[9] "Raspberry pi" – Google Trend [WWW] http://www.google.com/trends/explore#q=%22Raspberry%20Pi%22&cmpt=q (18.05.13)

[10] Garziano, Dan. „Raspberry Pi Linux PC orders now shipping“ [WWW] http://bgr.com/2012/04/16/raspberry-pi-linux-pc-orders-now-shipping/ (19.05.13)

[11] Garziano, Dan. „Raspberry Pi sales soar past 1 million as U.K production tops 500,000 units“ [WWW] http://bgr.com/2013/04/12/raspberry-pi-production-500000-units-433341/ (19.05.13)

[12] FAQ. – Raspberry Pi [WWW] http://www.raspberrypi.org/faqs (19.05.13)

[13] Broadcom. High Definition 1080p Embedded Multimedia Applications Processor - BCM2835. [WWW] http://www.broadcom.com/products/BCM2835 (01.06.13)

[14] ARM. ARM 1176 Processor. [WWW] http://www.arm.com/products/processors/classic/arm11/arm1176.php (01.06.13)

[15] „Universal Serial Bus Specification Revision 2.0“, USB Implementers Forum, 2000-04-27.

[16] „High Definition Multimedia Interface Specification Version 1.3“, HDMI Licensing, LLC, 2006-11-10.

[17] Hollenbeck, Rion. „The IEEE 802.3 Standard (Ethernet): An Overview of the Technology“, 2001-09-17 [WWW] http://rionhollenbeck.com/GradPortfolio/Papers/620-Ethernet/Ethernet.pdf (26.05.13)

[18] GPIO Interfaces. [WWW] https://www.kernel.org/doc/Documentation/gpio.txt (19.05.13)

[19] RPi Low-level peripherals. [WWW] http://elinux.org/Rpi_Low-level_peripherals (19.05.13)

[20] Pin Numbering. – Usage [WWW] http://pi4j.com/usage.html (19.05.13)

[21] RPi Hardware. [WWW] http://elinux.org/RPi_Hardware (19.05.13)

[22] SPI – Serial Peripheral Interface. - MCT [WWW] http://www.mct.net/faq/spi.html (26.05.13)

[23] „Overview of the Inter-Integrated Circuit (I^2C) BUS“ [WWW] http://ftp.utcluj.ro/pub/users/nedevschi/PMP/WLab/serial/i2c/i2c_over.pdf (26.05.13)

[24] Serial and UART Tutorial. - FreeBSD [WWW] http://www.freebsd.org/doc/en/articles/serial-uart/ (26.05.13)

[25] E-Teatmik. – Heikki Vallaste [WWW] http://www.vallaste.ee/index.htm?Type=UserId&otsing=796 (19.05.13)

[26] Temperature Data Loggers. – MicroDAQ [WWW] http://www.microdaq.com/data-logger/temperature/1.php (09.05.13)

[27] 3M Temperature Loggers. – 3M [WWW] http://solutions.3m.com/wps/portal/3M/en_EU/Temperature-Logger/Monitor-Shipments/ (09.05.13)

[28] LogTag Analyzer. – LogTag Recorders [WWW] http://www.logtagrecorders.com/software/software.htm (09.05.13)

[29] 9936A LogWare III. – Fluke Calibration [WWW] http://us.flukecal.com/products/calibration-software/temperature-calibration-software/9936a-logware-iii?geoip=1 (09.05.13)

[30] LogWare III User Guide revision: 850701. – Fluke Calibration [WWW] http://us.flukecal.com/products/calibration-software/temperature-calibration-software/9936a-logware-iii?quicktabs_product_details=4 (09.05.2013)

[31] Web-based temperature monitoring. iSD-TC. – OMEGA [WWW] http://www.omega.com/ppt/pptsc.asp?ref=ISD-TC (09.05.2013)

[32] User’s Guide. OMEGA® iSD-TC – Dual Thermocouple over the Internet [WWW] http://www.omega.com/Manuals/manualpdf/M4823.pdf (09.05.2013)

[33] „IEEE Recommended Practice for Software Requirements Specifications" *IEEE Std 830-1998,* Oct. 20 1998. [*Online*] IEEE Xplore (19.05.13)

[34] RPi Programming [WWW] http://elinux.org/RPi_Programming (19.05.13)

[35] wxPython [WWW] http://www.wxpython.org/ (19.05.13)

[36] QT - Product. [WWW] http://qt.digia.com/product/ (19.05.13)

[37] Sletta, Gunnar. „Introducing Boot to QT – A Technology Preview.“ – Qt Blog [WWW] http://blog.qt.digia.com/blog/2013/05/21/introducing-boot-to-qt-a-technology-preview/ (27.05.13)

[38] Extend QT Creator. – Qt Project [WWW] http://qt-project.org/wiki/Qt_Creator_Plug-in_Gallery (27.05.13)

[39] Creating Your First Plugin. – Qt Reference Documentation [WWW] http://doc.qt.digia.com/qtcreator-extending/first-plugin.html (27.05.13)

[40] Qwt User’s Guide. [WWW] http://qwt.sourceforge.net/ (20.05.13)

[41] QtCreator-Doxygen. Overview. [WWW] http://dev.kofee.org/projects/qtcreator-doxygen/ (27.05.13)

[42] Ubuntu QTCreator Plugins. [WWW] https://launchpad.net/ubuntu-qtcreator-plugins (27.05.13)

[43] 5 Steps for Selecting the Right USB to Serial adapter. - USConverters [WWW] http://www.usconverters.com/index.php?main_page=page&id=62&chapter=0 (19.05.13)

[44] Texas Instruments. LM50 SOT-23 Single-Supply Centigrade Temperature Sensor. [WWW] http://www.ti.com/lit/ds/symlink/lm50.pdf (30.05.13)

[45] Raspbian. [WWW] http://www.raspbian.org/ (19.05.13)

[46] QT Project. [WWW] http://qt-project.org/ (19.05.13)

[47] Docs – Qt 4.8. – Qt Project [WWW] http://qt-project.org/doc/qt-4.8/ (20.05.13)

[48] QFile Class Reference. – Qt Project [WWW] http://qt-project.org/doc/qt-4.8/qfile.html (20.05.13)

[49] QTextStream Class Reference. – Qt Project [WWW] http://qt-project.org/doc/qt-4.8/qtextstream.html (20.05.13)

[50] QDateTime Class Reference. – Qt Project [WWW] http://qt-project.org/doc/qt-4.8/qdatetime.html (20.05.13)

[51] QSettings Class Reference. – Qt Project [WWW] http://qt-project.org/doc/qt-4.8/qsettings.html (20.05.13)

[52] QMainWindow Class Reference. – Qt Project [WWW] http://qt-project.org/doc/qt-4.8/qmainwindow.html (20.05.13)

[53] Charles W. Krueger. Software reuse. – *Journal – ACM Computing Surveys (CSUR)*. June 1992, 24 (2), 131-183. [*Online*] ACM Digital Library (26.05.13)

[54] Horowitz, E.; Munson, John B., "An Expansive View of Reusable Software," *Software Engineering, IEEE Transactions on* , September 1984, SE-10 (5), 477 – 487. [*Online*] IEEE Xplore (20.05.13)

[55] qextserialport. [WWW] http://code.google.com/p/qextserialport/ (20.05.13)

[56] libqxt. [WWW] http://dev.libqxt.org/libqxt/wiki/Home (20.05.13)

[57] Qxt’s Modules. [WWW] http://libqxt.bitbucket.org/doc/tip/modules.html (20.05.13)

[58] QxtSmtp Class Reference. [WWW] http://libqxt.bitbucket.org/doc/tip/qxtsmtp.html (21.05.13)

[59] QxtMailMessage Class Reference. [WWW] http://libqxt.bitbucket.org/doc/tip/qxtmailmessage.html (21.05.13)

[60] QxtMailAttachment Class Reference. [WWW] http://libqxt.bitbucket.org/doc/tip/qxtmailattachment.html (21.05.13)

[61] GNU Octave. [WWW] http://www.gnu.org/software/octave/ (21.05.13)

[62] Polynomial Interpolation. [WWW] http://www.gnu.org/software/octave/doc/interpreter/Polynomial-Interpolation.html (21.05.13)

[63] The Method of Least Squares. - Efunda [WWW] http://www.efunda.com/math/leastsquares/leastsquares.cfm (27.05.13)

[64] Calling a Function by its Name. [WWW] http://www.gnu.org/software/octave/doc/interpreter/Calling-a-Function-by-its-Name.html (21.05.13)

[65] Introducing WinSCP. - WinSCP [WWW] http://winscp.net/eng/docs/introduction (23.05.13)

[66] Finajev, J. MATLAB/Simulink Compatible Heater Lab Kit with USB Interface : magistritöö. Tallinna Tehnikaülikool, Tallinn 2013.