

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
KOMPIUTERIJOS KATEDRA

Kursinis darbas

**Temperatūros stebėjimo sistemos kūrimas**

Atliko: 3 kurso studentai:

Tomas Novikovas 4gr

Martynas Raila 1gr

Darbo vadovas:

Rytis Malakauskas

Vilnius

2016

## Turinys

Anotacija .....	3
Summary .....	4
Įvadas .....	5
1. Reikalavimai .....	6
2. Sistemos komponentai .....	6
2.1. Naudojama įranga .....	6
2.2. Naudojamos sistemos.....	8
2.3. Programavimo kalba ir bibliotekos .....	8
3. Termometro parengimas .....	9
3.1. Patalpa .....	9
3.2. Kompiuterio ir sensorių sujungimas .....	9
3.3. Prisijungimas prie Raspberry Pi kompiuterio .....	10
4. Termometro programavimas .....	11
4.1. Sensorių informacijos skaitymas.....	11
4.2. Lokalus informacijos saugojimas.....	11
4.3. Pranešimų siuntimai .....	11
4.4. Nustatymų failas.....	12
4.5. Duomenų perdavimas į virtualią mašiną.....	13
5. Virtualios mašinos konfigūracija .....	14
5.1. Pagrindinių programinių paketų instaliacija .....	14
5.2. Duomenų gavimas iš Raspberry Pi .....	14
5.3. „Cacti“ konfigūracija .....	14
6. Diagramų kūrimas „Cacti“ sistemoje.....	15
7. Testavimas.....	16
Išvados ir ateities planai .....	17
Literatūros sąrašas .....	18

## **Anotacija**

Kursinio darbo užduotis yra sukurti Vilniaus universiteto matematikos ir informatikos fakulteto skaitmeninių tyrimų ir skaičiavimų centro (VU MIF STSC) superkompiuterio patalpos temperatūros bei oro drėgnumo matavimo sistemą. Termometrui kurti naudojamas Raspberry Pi kompiuteris. Sukurta sistema geba matuoti temperatūrą ir oro drėgnumą VU MIF STSC superkompiuterio patalpoje. Temperatūra ir oro drėgnumas yra diagramų pagalba pavaizduojamas „Cacti“ sistemoje, kurią universitetas naudoja serverių ir kitų įrenginių būsenai stebėti. Taip pat sistema turi galimybę siųsti elektroninius laiškus, kai sugenda vienas iš sensorių arba temperatūra ar oro drėgnumas patalpoje peržengia numatytas ribas.

## **Summary**

### **Temperature monitoring system development**

Coursework assignment is to create the VU MIF STSC supercomputer room temperature and air humidity measurement system. A Raspberry Pi computer is used to create the thermometer. The developed system is able to measure the temperature and air humidity in the VU MIF STSC supercomputer room. Temperature and humidity are displayed via graphs in the Cacti system, which University uses to track use of servers and the status of other equipment. The system also has the ability to send e-mails when one of the sensors fails or the temperature or humidity in the room goes beyond the limits imposed.

## Ivadas

Projekto užduotis yra sukurti Vilniaus universiteto matematikos ir informatikos fakulteto skaitmeninių tyrimų ir skaičiavimų centro superkompiuterio patalpos temperatūros bei drėgnumo matavimui skirtą termometrą. Šis termometras bus naudojamas VU MIF STSC darbuotojų, kurie atsakingi už superkompiuterio veikimą. Termometras taip pat turi būti suderinamas su jau veikiančia „Cacti“ sistema, kurią naudoja VU MIF darbuotojai. Ši sistema yra skirta stebėti įvairių universitete esančių serverių ar kitų kompiuterinių prietaisų apkrovą, būseną, temperatūrą ir panašiai.

Prieš kuriant sistemą temperatūra ir oro drėgnumas VU MIF superkompiuterio patalpoje buvo matuojami su paprastu skaitmeniniu termometru, kuris yra pritvirtintas priešais kamerą. Šis matavimo būdas nėra pats geriausias, nes nuolat reikalauja šviesos. Jei patalpoje tamsu – termometro ekrane nieko nesimatys. Be to, neįmanoma stebėti kaip keičiasi temperatūra ar oro drėgnumas patalpoje laikui bėgant. Matoma tik reikšmė esamu laiku. Taip pat termometras užstoja dalį vaizdo, kurį stebi vaizdo kamera.

Su šiuo, jau sukurtu termometru temperatūrą ir oro drėgnumą bei jų pokyčius galima stebėti „Cacti“ sistemoje. Termometras taip pat geba siųsti pranešimus į nurodytus elektroninio pašto adresus, jei vienas iš sensorių sugenda arba oro temperatūra ar drėgnumas patalpoje pakyla arba nusileidžia per daug. Be to, termometras yra nesunkiai praplečiamas. Tai yra, nesunkiai pridedama daugiau duomenis matuojančių sensorių, jei atsirastų noras oro temperatūrą ir drėgnumą matuoti ne tik superkompiuterio patalpoje, bet ir kitose vietose.

Šio aprašo tikslas yra aprašyti sukurtą termometrą, bei jo funkcionalumą. Pirmame skyriuje aprašyti užduoties reikalavimai. Antrame skyriuje aprašyti sistemoje naudojami komponentai. Trečiame skyriuje aprašytas termometro konstravimas, bei sujungimas. Ketvirtame skyriuje aprašytas programos, naudojamos gauti duomenis iš sensorių kūrimas. Penktame skyriuje aprašytas virtualios mašinos ir joje esančios „Cacti“ sistemos konfigūravimas. Šeštame skyriuje aprašyta, kaip naudojant „Cacti“ sistemą yra kuriamos diagramos. Septintame skyriuje aprašytas visos sukurtos sistemos testavimas.

## **1. Reikalavimai**

Pagrindiniai kursinio darbo reikalavimai yra du. Sistema turi skaityti duomenis apie oro temperatūrą ir drėgnį superkompiuterio patalpoje ir juos atvaizduoti sistemoje, kurią naudoja Vilniaus universiteto matematikos ir informatikos fakulteto skaitmeninių tyrimų ir skaičiavimų centro darbuotojai.

Kiti reikalavimai kursiniam darbui:

- sistema turi rinkti duomenis bent iš 4 sensorių (2 šaltojoje ir 2 karštojoje zonoje)
- sistema turi būti nesunkiai plečiama. Tai yra, neturi būti sunku pridėti daugiau duomenų rinkimo taškų arba papildomų funkcijų
- sistema turi turėti galimybę pranešti darbuotojams apie neveikiančius sensorius arba staigius temperatūros pokyčius

## **2. Sistemos komponentai**

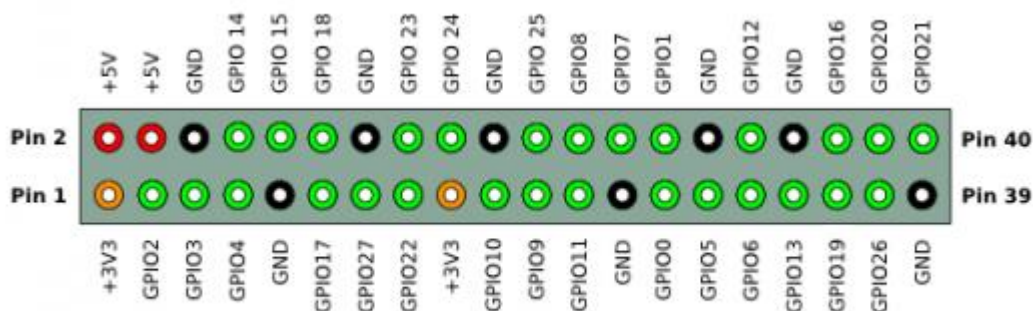
Sistema sudaryta iš nemažai skirtingų komponentų. Taip yra dėl to, kad kursinio darbo užduotis yra ne tik suprogramuoti termometrą superkompiuterio patalpai, bet ir jį surinkti. Termometro surinkimui reikalingi sensoriai, skaitantys oro temperatūros bei drėgnumo duomenis, reikalingas kompiuteris, kuris apdoro duomenis gaunamus iš sensorių, reikalingi laidai, kuriais sensoriai bus jungiami prie kompiuterio. Taip pat norint atvaizduoti duomenis grafikuose bus reikalinga speciali programinė įranga. Naudojami sistemos komponentai plačiau aprašyti šiame skyriuje.

### **2.1. Naudojama įranga**

Termometrui naudojamas Raspberry Pi 2 Model B kompiuteris. Jame įdiegta „Raspbian“ operacinė sistema. „Raspbian“ yra paremta „Debian“ operacine sistema optimizuota veikti su „Raspberry“ programine įranga.

Raspberry Pi 2 turi keturiasdešimt bendros paskirties įvesties ir išvesties jungčių, toliau GPIO, (GPIO – General Purpose Input Output), kurios naudojamos kaip sąsaja tarp kompiuterio ir kitų įrenginių [2]. Iš keturiasdešimties jungčių (žr. 1 Pav.) dvi yra skirtos penkių voltų elektros srovės

įtampai (raudona spalva), dvi – trijų voltų įtampai (oranžinė spalva), aštuonios jungtys – įžeminimui (juoda spalva) ir likusios 28 – įvesties ir išvesties duomenims siųsti (žalia spalva).



**1 Pav. Bendros paskirties įvesties ir išvesties jungtys**

Bendros paskirties įvesties ir išvesties jungtys yra numeruojamos dviem būdais: eilės numeriu ir jungties numeriu. Eilės numeriu yra numeruojamos visos jungtys, jungties numeriu tik jungtys skirtos siųsti arba gauti duomenis (žalia spalva). Visame apraše jungtys bus numeruojamos jungties numeriais, nes įtampos bei įžeminimo jungčių numeriai neturi reikšmės ir programos kode yra naudojami tik jungčių (GPIO) numeriai.

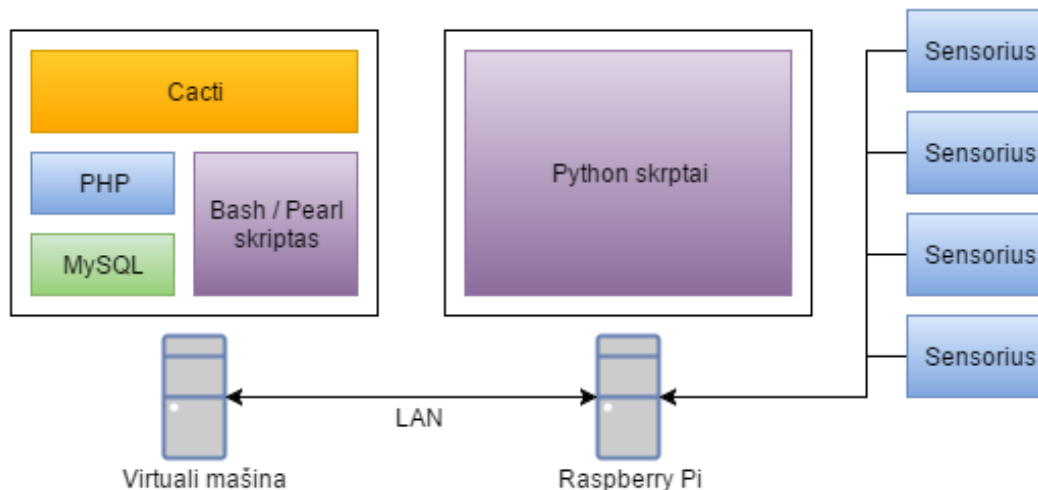
Temperatūrai ir drėgnumui matuoti naudojami „AM2302/DHT22“ sensoriai [1]. Sensoriai yra pritaikyti naudoti su 3 voltų ir 5 voltų elektros srovės įtampa. Maksimalus atstumas, kuriuo sensorius veikia yra šimtas metrų. Dažniausias intervalas, kuriuo gali būti renkami duomenys – dvi sekundes. Šie sensoriai pasirinkti dėl to, kad šių sensorių gavome iš projekto vadovo, jų funkcionalumo pilnai užtenka įgyvendinti projektą, yra pigūs bei turi biblioteką programavimui.

Sensorius turi keturias jungtis. Pirma skirta tiekti elektros srovę į sensorių, antra – duomenims gauti, o trečia ir ketvirta – įžeminimui. Nors sensorius turi dvi įžeminimo jungtis, užtenka naudoti tik vieną. Todėl projekto metu sensorių įžeminimui naudosime ketvirtą jungtį.

Sensorių jungimui prie kompiuterio naudojami telefoniniai trečios kategorijos (cat-3) keturių gyslų kabeliai. Tokie kabeliai pasirinkti dėl to, kad juos gauname iš projekto vadovo ir jų pilnai užtenka įgyvendinti projektą, kadangi sujungti sensorius užtenka trijų laido gijų. Naudojant UTP interneto kabelius būtų neišnaudotos penkios laido gijos.

## 2.2. Naudojamos sistemos

Mūsų projektą sudaro dvi sistemos (žr. 2 Pav.). Pirmoji – Raspberry Pi naudojanti termometro sistema, kuri surenka duomenis iš prie jos prijungtų matavimo sensorių. Antroji – virtuali mašina, kurioje veikia „Cacti“ sistema, priimanti duomenis iš pirmosios sistemos bei juos atvaizduojanti diagramose.



2 Pav. Naudojamų sistemų schema

Raspberry Pi sistemoje yra laikomos Python programa, skirta nuskaityti duomenis iš prijungtų sensorių. Virtualioje mašinoje yra įrašyta „Cacti“ sistema, bei jai reikalingos PHP bibliotekos, taip pat „MySQL“ duomenų bazės valdymo sistema. Virtualioje mašinoje taip pat yra programa skirta gauti ir formatuoti duomenis gautus iš Raspberry Pi kompiuterio.

## 2.3. Programavimo kalba ir bibliotekos

Termometrui kurti naudojama „Python“ programavimo kalba. Ji pasirinkta dėl to, kad oficialius sensorių gamintojas „Adafruit“ pateikia biblioteką, parašyta šia kalba.

Bendrauti su sensoriais naudojama papildoma biblioteka „Adafruit\_Python\_DHT“. Tai biblioteka, sukurta specialiai DHT serijos sensoriams naudojamiems su „Raspberry Pi“. Biblioteka pritaikyta „Python 2.6/2.7“ versijai. Biblioteka yra nemokama. Licencija leidžia biblioteką bei su biblioteka susijusius failus naudoti, keisti, parduoti ar licencijuoti.

Pagrindiniai naudojami bibliotekos metodai yra `read` ir `read_retry`. Tai metodai, kurie gauna sensoriaus tipą, jei jungties numerį ir grąžina iš sensoriaus nuskaitytą kortežą (angl. Tuple) su oro drėgnumu ir temperatūra. Skirtumas tarp metodų tas, kad metodas „`read`“ iš sensoriaus duomenis skaito tik vieną kartą ir grąžina nuskaitytą kortežą, o metodas „`read_retry`“ duomenis skaito penkiolika kartų ir grąžina pirmą nuskaitytą netuščią kortežą. Tokie du beveik vienodi metodai yra naudojami dėl to, kad pasitaiko atveju, kai iš pirmo karto nepavyksta gauti sensorių nuskaitytos informacijos su temperatūra ir oro drėgnumu ypatingai, kai sensorius yra sujungtas ilgesniais laidais. Informacijos iš sensoriaus pirmu kartu gauti nepavyksta dažniausiai dėl to, kad baigiasi atsakymo iš sensoriaus laukimo laikas.



### **3. Termometro parengimas**

Viena iš užduočių kursinio darbo metu yra ne tik suprogramuoti termometrą, kuris matuotų oro temperatūrą ir oro drėgnumą VU MIF superkompiuterio patalpoje, bet ir sukonstruoti patį termometrą ir paruošti jį naudojimui. Termometras turi būti lengvai tobulinamas, t. y. turi būti nesunku pridėti papildomų dalių ar papildomų sensorių. Termometras vienu metu daugiausiai gali turėti keturis aktyvius sensorius dėl Raspberry Pi jungčių, kurios teikia elektros srovę sensoriams, kiekio. Norint su Raspberry Pi naudoti daugiau sensorių reikia juos jungti prie atskiro srovės šaltinio.

#### **3.1. Patalpa**

Patalpa, kurioje stovi VU MIF superkompiuteris yra apytiksliai 8 metrų ilgio ir 5 metrų pločio. Išilgai per vidurį patalpos stovi superkompiuteris ir visą patalpą padalina į dvi dalis, vadinamas karštąja ir šaltąja zonomis.

Karštoji zona yra ta pusė, į kurią išpučiamas šiltas oras iš superkompiuterio. Ši superkompiuterio pusė yra šiltesnė ir oro temperatūra yra šiek tiek aukštesnė, apie 20-24°C. Šaltoji zona yra ta pusė, kurioje superkompiuterio vėsinimui paduodamas šaltas oras ir kondicionierių. Temperatūra šioje zonoje siekia apie 14-19°C.

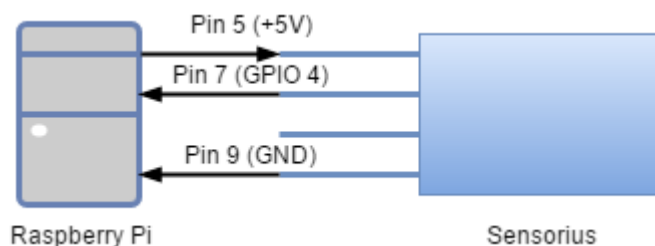
#### **3.2. Kompiuterio ir sensorių sujungimas**

Sensoriai pastatyti keturiuose patalpos kampuose prie lubų. Du šaltojoje ir du karštojoje zonose. Kompiuteris, prie kurio laidais jungiasi sensoriai yra užkeltas virš pakabinamų lubų, netoli vieno sensoriaus karštojoje zonoje. Atstumai nuo kompiuterio iki sensorių yra apytiksliai: karštojoje zonoje – 1 metras ir 8 metrai, šaltojoje zonoje – 4 metrai ir 8 metrai. Žinoma laidai, skirti sujungti sensorius su kompiuteriu yra ilgesni, nes ne visi laidai nutiesti tiesiai iki sensoriaus, taip pat šiek tiek laido yra palikta atsargai.

Sensorių sujungti su kompiuteriu naudojame ilgus telefoninius 3 kategorijos kabelius ir trumpus pereinamus laidus su antgaliais, kurie tiek kompiuterio, tiek sensorių jungtis jungia su laidu. Sensoriui sujungti reikalingos trys laido gijos, kurie jungiami į tam skirtas Raspberry Pi jungtis ir į teisingas jungtis sensoriuje.

3 Pav. pavaizduotas pavyzdinis vieno sensoriaus sujungimas su Raspberry Pi kompiuteriu. Sensorius turi keturias jungtis, tačiau naudojamos yra tik trys. Juodomis rodyklėmis pavaizduoti laidai, kurie yra naudojami jungiant sensorių. Pirmoji sensoriaus jungtis jungiama su 5 jungtimi Raspberry Pi kompiuteryje. Penkta jungtis yra skirta 5 voltų elektros srovei tiekti. Antroji sensoriaus jungtis yra jungiama su 7 jungtimi kompiuteryje. Ši jungtis yra naudojama informacijos gavimui iš sensoriaus. Programoje yra naudojamas GPIO numeravimas, todėl ši jungtis

identifikuojama kaip GPIO4. Trečioji sensoriaus jungtis nėra naudojama. Ketvirtoji sensoriaus jungtis jungiama su 9 jungtimi kompiuteryje. Ši jungtis naudojama sensoriaus įžeminimui.



**3 Pav. Pavyzdinis vieno sensoriaus sujungimas su Raspberry Pi**

Sensoriui sujungti naudojamos trys telefoninio kabelio gijos. Viena elektros srovei, antra duomenims siųsti ir trečia – įžeminimui. Sutarta, kad kiekvienai elektros srovės gijai naudojame oranžinės spalvos gijas, duomenims siųsti – mėlynos spalvos gijas, o įžeminimui – oranžinės su balta spalvos gijas. Tokiu būdu nesunku susigaudyti, kokią funkciją atlieka kiekviena laido gija. Trys skirtingos spalvos naudojamos dėl to, kad ilgais atstumais siunčiant duomenis tuo pačiu laidu kaip ir elektros srovę, informacija gali būti pakitusi dėl susidariusių elektromagnetinių laukų.

### **3.3. Prisijungimas prie Raspberry Pi kompiuterio**

Prisijungti prie Raspberry Pi kompiuterio, stovinčio VU MIF superkompiuterio patalpoje, galima dviem pagrindiniais būdais. Pirmas būdas prisijungti prie kompiuterio yra naudojant monitorių ir klaviatūrą. Tačiau taip daryti nepatogu, nes kompiuteris guli gana aukštai. Antras būdas – prisijungti prie kompiuterio nuotoliniu būdu naudojant SSH (Secure Shell) protokolą. Raspberry Pi kompiuteris savo programinėje įrangoje turi pagal nutylėjimą įjungtą SSH serverį. Prisijungti prie kompiuterio galima tik naudojant kitą kompiuterį, esantį tame pačiame tinkle. Tokiu būdu prisijungus prie Raspberry Pi kompiuterio galima naudotis tik komandine eilute, tačiau to pilnai užtenka.

## **4. Termometro programavimas**

Kita užduotis jau sukonstravus termometrą yra sukurti programą, kuri gebėtų nuskaityti prijungtus sensorius, apdoroti informaciją. Šiame skyriuje yra aprašytas programos, esančios Raspberry Pi kompiuteryje funkcionalumas. Programa gali perskaityti informaciją iš prijungtų sensorių, saugoti informaciją lokaliai ir siųsti pranešimus esant tam tikroms sąlygoms. Taip pat programa turi nustatymų failą, kuris yra naudojamas programoje.

### **4.1. Sensorių informacijos skaitymas**

Informacija iš sensorių yra skaitoma naudojant jau aprašytą Python programavimo kalbos biblioteką. Siekiant programą padaryti kuo labiau dinamišką (lengviau pildomą ir plečiamą), visi naudojami Raspberry Pi jungčių numeriai (GPIO numeriai) yra surašyti į nustatymų failą. Programa nuskaityti visus jungčių numerius, patikrina ar numeriai tikrai yra visų įmanomų GPIO jungčių sąraše ir juos sudeda į sąrašą. Ciklo pagalba yra nuskaityta kiekviena jungtis ir informacija yra surašoma į vieną žodyną (angl. dictionary) tolimesniam informacijos naudojimui.

### **4.2. Lokalus informacijos saugojimas**

Kiekvieną kartą, kai programa yra paleidžiama ir nuskaityti duomenys jie yra lokaliai saugomi Raspberry Pi. Metodas panaudoja žodyną su visomis temperatūros ir oro drėgnumo reikšmėmis ir šiuos duomenis išsaugo faile.

Siekiant išvengti labai didelių failų nuspręsta kiekvieną dieną kurti po naują failą saugoti duomenims. Failo pavadinimas – pilna dienos data: metai, mėnesis ir diena. Tai padaroma kiekvieną kartą prieš rašant duomenis tikrinant sistemos datą. Jei failas su esama data jau yra sukurtas, tai duomenys tiesiog įrašomi į failo apačią. Jei failas neegzistuoja, tai prieš rašant yra sukuriamas.

Kadangi programa privalo būti leidžiama su administratoriaus „root“ teisėmis, tai ir programos metu sukurti failai priklauso administratoriui. Jei failas paleidžiamas kito vartotojo, vartotojas sukurtą failą su saugomais duomenimis peržiūrėti negali. To išvengiama kode po failo sukūrimo pakeičiant failo savininkus į vartotoją, kuriam priklauso direktorija, iš kurio programa paleidžiama.

Informacijos saugojimas atliekamas kas penkias minutes pasinaudojant „cron“ darbu. Jis paleidžia programą ir įrašo sensorių duomenis į failą. Šis failas yra naudojamas „Cacti“ sistemos, kaip duomenų šaltinis. Kai „Cacti“ sistema kreipiasi į Raspberry Pi, iš šio failo yra nuskaityta paskutinė duomenų eilutė.

### **4.3. Pranešimų siuntimai**

Programa taip pat turi funkciją siųsti elektroninius laiškus apie termometro veikimą. Šiuo metu pranešimai yra siunčiami tuo atveju, kai nepavyksta gauti duomenų iš nurodyto sensoriaus arba

kai viena iš nuskaitytų reikšmių yra didesnė negu numatytos minimalios ir maksimalios temperatūros bei oro drėgnumo reikšmės.

Tai padaroma tikrinant jau užpildytą žodyną su temperatūros bei drėgnumo reikšmėmis. Jei nors viena reikšmė yra be duomenų arba didesnė / mažesnė už numatytus rėžius – elektroninis laiškas yra siunčiamas. Minimali leidžiama temperatūra yra 10 laipsnių, maksimali – 30 laipsnių. Minimalus oro drėgnumas yra 30 procentų, o maksimalus – 70 procentų. Minimalios ir maksimalios reikšmės nustatytos naudojant apytikslius duomenis testavimo metu ir naudojant apytiksles kondicionierių, stovinčių VU MIF superkompiuterio patalpoje reikšmes.

Elektroniniai laišakai yra siunčiami naudojant Gmail elektroninį paštą. Prieš siuntimą žinutė yra suformatuojama, surašomos sensorių reikšmės ir į siuntėjo adreso laukelį įrašomas sukurtos Gmail paskyros adresas. Elektroninių laiškų gavėjai yra nuskaityti iš nustatymų failo.

Viena iš idėjų kursinio darbo metu buvo siųsti ne tik elektroninius laiškus, bet ir trumpąsias SMS žinutes, naudojant internetines žinutes. Šios idėjos nepavyko įgyvendinti, nes daugelis internetinių SMS žinučių tiekėjų paslaugos Lietuvoje neteikia, jos yra mokamos arba prie žinučių pridamos reklamos. Be to, nusprendėme neviešinti telefono numerių, nes yra galimybė, kad numerius pradės naudoti reklamuoti prekes ar paslaugas.

Raspberry Pi turi galimybę įdiegti modulį su SIM kortele. Tai leistų siųsti SMS žinutes, tačiau ši galimybė buvo apsvastyta per vėlai ir atsisakyta dėl laiko stokos.

#### **4.4. Nustatymų failas**

Visi nustatymai ir duomenys, reikalingi gauti duomenis iš sensorių ir elektroninio pašto adresai yra laikomi nustatymų faile. Paleidus programą patikrinama ar failas yra parašytas be klaidų, nuskaityti reikiami duomenys ir vėliau naudojami programoje.

Pirma idėja buvo nenaudoti nustatymų failo ir visus naudojamus duomenis nurodyti komandinėje eilutėje paleidžiant programą. Tačiau kaskart leidžiant programą reikėtų nurodyti visus sensorius, kurių gali būti daug, taip pat teisingai nurodyti visus elektroninio pašto adresus. Tai yra nepatogu ir sunku pakeisti ar papildyti. Nustatymų failas leidžia lengviau keisti ir pildyti naudojamus duomenis.

Norint paleisti programą, failas turi būti suformatuotas teisingai. Nustatymų faile saugomi trys reikalingi nustatymai, kurie turi prasidėti teisingais žodžiais:

- Sensoriaus tipas. Turi prasidėti žodžiu: „type:“
- Naudojamos GPIO jungtys. Turi prasidėti žodžiu: „gpio:“
- Elektroninio pašto adresai. Turi prasidėti žodžiu: „email:“

Po dvitaškių kiekvienoje eilutėje yra nurodomi duomenys turi būti atskiriami tarpais ir negali būti tušti. Išimtis yra sensoriaus tipas. Jei nenurodytas joks tipas, bus programoje bus naudojamas

2302 sensorių tipas. Visos tuščios eilutės ir eilutės, kurios prasideda su grotelių ženklu (#) yra ignoruojamos.

4 Pav. pavaizduotas pavyzdinis nustatymų failas. 1, 4, 7 eilutės bus ignoruojamos, nes jos prasideda grotelių ženklu. 3, 6, 9 eilutės taip pat bus ignoruojamos, nes jos tuščios. 2 eilutėje nurodytas naudojamų sensorių tipas. Kadangi tipas nenurodytas, programa automatiškai naudos AM2302 sensorius. 5 eilutėje po žodžio „GPIO:“ nurodytos visos jungtys, iš kurių programa skaitys duomenis. 8 eilutėje po žodžio „email:“ nurodyti visi elektroninio pašto adresai, į kuriuos bus siunčiami pranešimai, kai neveikia nors vienas sensorius arba viena iš nusikaitytų reikšmių yra per maža arba per didelė.

```
1 # Sensor type. 11 22 or 2302
2 type:
3
4 # Used GPIOs (separate with spaces)
5 GPIO: 4 5 6 7
6
7 # Email addresses (separate with spaces). Used to send errors
8 email: example@gmail.com example@example.lt
9
```

4 Pav. Pavyzdinis nustatymų failas

Ištrynus nustatymų failą ir kitą kartą leidžiant programą automatiškai yra sugeneruojamas naujas nustatymų failas, kuriame reikės prie atitinkamų eilučių įrašyti norimus duomenis. Kitu atveju programa nepsileis ir duomenų iš sensorių gauti nebus įmanoma.

#### 4.5. Duomenų perdavimas į virtualią mašiną

Raspberry Pi įrašyti SNMP (angl. Simple Network Management Protocol) ir SNMPD programų paketai. SNMP konfigūracijos faile pridėti papildomi parametrai, kuriuose buvo nurodytas IP adresas, į kurį bus siunčiami duomenys (Virtualios mašinos IP adresas), bei komanda su unikaliu OID numeriu, kuris paleis „Python“ kodą sistemoje, gražinantį duomenis iš sensorių. OID yra unikalus objektą identifikuojantis numeris, naudojamas atskirti SNMP žinutes. Pasinaudojus „snmpwalk“ funkcija bei komandai priskirtu OID numeriu, terminale atrenkamas vienas tikslus OID, kuris gražina reikalingus duomenis. Šis OID naudojamas gauti sensorių duomenims iš Raspberry PI virtualios mašinos „Bash / Perl“ skripte.

## **5. Virtualios mašinos konfigūracija**

Sekanti užduotis yra sukongfigūruoti bei paruošti virtualią mašiną, kad ji, naudodama Raspberry Pi sensorių duomenis, galėtų atvaizduoti juos diagramų pagalba. Pirmiausia sistemoje įrašomi baziniai programų paketai. Taip pat sukuriamas skriptas, kuris bus reikalingas „Cacti“ sistemai, kad ji galėtų gauti duomenis iš Raspberry Pi. Pabaigoje aprašoma „Cacti“ sistemos instaliacija pasinaudojant naršykle [3].

### **5.1. Pagrindinių programinių paketų instaliacija**

Virtualioje mašinoje pirmiausia buvo įrašytas „cacti-spine“ programų paketas. Jis įrašo visus papildomus programų paketus reikalingus funkcionaliam „Cacti“ darbui („apache2“, „PHP 7“, „MySQL 5.2“). Instaliacija turi patogų įrašymo procesą ir yra intuityvi, leidžianti sukurti visus reikalingus vartotojo prisijungimus prie programų instaliacijos procese. Taip pat įrašyti SNMP ir SNMPD programų paketai, leidžiantys gauti bei siųsti informacija SNMP protokolu. SNMP konfigūracijos faile buvo pridėta papildomi parametrai, kuriuose buvo nurodyta IP adresas, iš kurio bus gaunami duomenys (Raspberry Pi IP adresas).

### **5.2. Duomenų gavimas iš Raspberry Pi**

Virtualioje mašinoje taip pat buvo sukurtas „Bash / Perl“ skriptas, kuris kas penkias minutes kreipiasi į Raspberry Pi per SNMP protokolą. Pasinaudojus „snmpget“ funkcija bei unikaliu OID, gaunami duomenys iš sensorių. Duomenys po to yra suformatuojami „Perl“ kodo pagalba, tokiu formatu: „pavadinimas:vertė pavadinimas:vertė pavadinimas:vertė ir t.t.“. Šioje vietoje „Perl“ kodas pasirūpina, kad duomenų formatas būtų priimtinas „Cacti“ sistemai.

### **5.3. „Cacti“ konfigūracija**

Virtualios mašinos naršyklėje įvedus „http://(virtualios mašinos IP adresas)/cacti“, vartotojas pasiekia internetinę „Cacti“ instaliaciją, kurioje sistema parodo reikalingus ir įrašytus programų paketus, bei leidžia pasirinkti, kokias „Cacti“ komponentų versijas norėsime naudoti. Pabaigus instaliaciją, prie „Cacti“ sistemos galima prisijungti suvedus atitinkamus prisijungimo duomenis. Pirmą kartą prisijungiant reikia įvesti iš anksto numatytus vartotojo duomenis (vartotojas: admin, slaptažodis: admin). Po prisijungimo vartotojas yra priverstas pasikeisti savo slaptažodį.

## 6. Diagramų kūrimas „Cacti“ sistemoje

Šiame skyriuje aprašytas procesas, reikalingas tam, kad „Cacti“ sistemoje būtų sukurta diagrama, pasinaudojus gautais duomenimis [3].

Pirmiausia sukuriamas naujas datos įvedimo metodas (angl. Data Input Methods). Norint tai padaryti reikia užpildyti pavadinimo, pasirinktos įvesties tipo ir sistemos kelio iki skripto, kuris gražina duomenis, laukus. Norint perduoti papildomus parametrus skriptui – reikia prie kelio iki failo pridėti papildomus kintamuosius. Nurodžius papildomus parametrus reikia kiekvienam iš jų pridėti papildoma įvedimo laukelį (angl. Input Field). Jam suteikti pavadinimą ir „draugišką“ pavadinimą. Kiekvienai mūsų gaunamai duomenų „pavadinimas:vertė“ porai sukuriame po išvesties laukelį (angl. Output Field) su pavadinimu ir „draugišku“ pavadinimu.

Toliau konfigūruojant „Cacti“ sistemą yra sukuriamas naujas prietaisas (angl. Device). Įvedamas prietaiso aprašymas/pavadinimas ir IP adresas. Pasirenkamas šeimininko šablonas (angl. Host Template), šiuo atveju „Local Linux Machine“. Papildomų parametrų keitimas priklauso nuo situacijos ir nėra būtinas.

Sekantis žingsnis sukurti duomenų šaltinį (angl. Data Source). Pasirenkamas šeimininkas (angl. Host) iš pateikiamo sąrašo. Duomenų šaltiniui suteikiamas pavadinimas. Pasirenkame duomenų įvedimos metodas iš esamo sąrašo. Kiekvienam esamam įvedimo laukeliui duomenų įvedimo metode priskiriama vertė. Kiekvienam išvedimo laukeliui sukuriamas naujas duomenų šaltinio elementas, kuriame įvedame vidinį duomenų šaltinio pavadinimą ir prie išvesties laukelio pasirenkame norimą duomenų porą, kuri priklausys šiam duomenų šaltinio elementui.

Atlikus šiuos žingsnius gali būti sukuriamos duomenis atvaizduojančios diagramos. Diagramų valdymo lange, galime sukurti naują diagramą arba keisti jau esamas. Norint pavaizduoti duomenis diagramoje, turime pridėti naują diagramos elementą. Jam iš esamo sąrašo pasirenkamas šeimininkas. Iš duomenų šaltinių sąrašo pasirenkami mus dominantys duomenys, kuriuos apsirašėme išvesties laukeliuose duomenų šaltinyje. Pasirenkamos duomenis atvaizduojančios spalvos, diagramos elemento tipai ir konsolidacijos funkcija.

## 7. Testavimas

Programa, nuskaitanti duomenis iš sensorių buvo testuojama prijungiant sensorius prie jungčių ir leidžiant programą per patį Raspberry Pi. Vėliau, kai Raspberry Pi buvo sujungtas su sensoriais ir pastatytas VU MIF STSC superkompiuterio patalpoje, testavimas buvo atliekamas jungiantis prie Raspberry Pi nuotoliniu būdu. Sensorių veikimas testuotas kartu su visa programa ir taip pat duomenis lyginant su esamu termometru superkompiuterio patalpoje.

„Cacti“ sistemos testavimas buvo atliktas virtualioje aplinkoje. Tam buvo pasitelkta virtualizacijos programa „VirtualBox“, kurios pagalba buvo sukurtos dvi virtualios mašinos su „Ubuntu 16.04“ operacinėmis sistemomis. Abi mašinos naudojo „bridged network“ tinklo adapterį, tam, kad gautų individualius vidinius IP adresus bei galėtų pasiekti internetą. Pirmoji mašina imitavo „Cacti“ serverį, o antroji – Raspberry Pi kompiuterį.

Taip pat nemažai laiko buvo skirta testuoti duomenų perdavimui per SNMP protokolą. Taip pat teisingai sukonfigūruoti virtualias mašinas, kad „Cacti“ serveris galėtų kreiptis į Raspberry Pi, o šis jam gražintų duomenis. Šioje vietoje padėjo „log“ failų stebėjimas ir pateiktos informacijos analizavimas.



## **Išvados ir ateities planai**

Mūsų sukurta sistema atitinka mūsų išsikeltus reikalavimus. Ji gauna duomenis iš keturių sensorių, esančių VU MIF skaitmeninių tyrimų ir skaičiavimų centro (STSC) superkompiuterio patalpoje, juos perduoda virtualiai mašinai, kurioje veikia „Cacti“ sistema. Ši sistema, pasinaudojus gautais duomenimis, atvaizduoja juos diagramų pavidalu.

Sistema gali būti patobulinta keliuose vietose. Pirmiausia galima padaryti, kad duomenų rinkimas Raspberry Pi sistemoje vyktų pasinaudojus duomenų baze. Gautus duomenis talpinti į ją, reikiamus įrašus ištraukti iš jos. Tai padėtų geriau archyvuoti duomenis, bet turėti patogesnę prieigą prie jų.

Taip pat galima atsisakyti visų tarpinių „Bash“ skriptų ir sukonfigūruoti „Cacti“ sistemą, kad ji tiesiogiai, pasinaudojus SNMP protokolu, gautų duomenis iš Raspberry Pi sistemos. Tai sumažintų „lūžio“ taškų kiekį ir padarytų sistemą patikimesnę.

Sistema veikia ir atlieka savo darbą. Esam patenkinti savo pasiektu rezultatu bei sukurta sistema. Šio projekto kūrimas mums suteikė naujų žinių bei patirties, dirbant su technologijomis, su kuriomis iki šiol nebuvo susidūrę. Tikimės, kad ši sistema bus naudojama ir tobulinama ateityje.

## Literatūros sąrašas

- [1] Digital relative humidity & temperature sensor AM2302/DHT22  
(<https://www.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf>)
- [2] RPi Low-level peripherals  
([http://elinux.org/RPi\\_Low-level\\_peripherals#Internal\\_Pull-Ups\\_.26\\_Pull-Downs](http://elinux.org/RPi_Low-level_peripherals#Internal_Pull-Ups_.26_Pull-Downs))
- [3] Cacti documentation  
(<http://www.cacti.net/documentation.php>)