VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Baigiamasis bakalauro darbas

Bičių avilių monitoringas

Atliko: 4 kurso, 4 grupės studentas
Jonas Gucu (parašas)

Darbo vadovas:
dr. Tomas Plankis

Vilnius
2017

Turinys

Įvadas **6**
1. Tradiciniai ir Langstroth‘o aviliai **8**
1.1. Tradiciniai aviliai 8
1.2. Langstroto avilys 8
1.3. Langstroth‘o avilio struktūra 9
2. Bičių avilių monitoringas **13**
2.1. Bičių avilių tikrinimas 13
2.2. Elektroninis bičių avilių monitoringas 13
2.3. Stebimi parametrai 14
3. Bičių avilio monitoringo sistemos apžvalga **16**
4. Arduino plokštė ir principinė schema **18**
4.1. Kas yra Arduino Uno? 18
4.2. Kodėl Arduino Uno? 18
4.3. Iš ko susideda Arduino programėlė? 18
4.4. Arduino plokštės ir daviklių pajungimas prie modulinio avilio 19
4.5. Temperatūros matavimas su Arduino Uno 20
5. Praktinis temperatūros matavimas **21**
5.1. Pasiruošimas darbui 21
5.2. Termistorius 21
5.3. Analoginis-skaitmeninis keitiklis 22
5.4. Įtampos daliklis 22
5.5. Termistoriaus grandinė 23
5.6. Eskizas 24
5.7 Temperatūros matavimas skaitmeniniu DS18B20 termistoriumi 28
5.8 Termistoriaus ir skaitmeninio termometro palyginimas 31
6. Avilio monitoringo sistema **33**
6.1 Arduino eskizas 34
6.2 Web aplikacija 35
6.3 Konsolinė aplikacija 37
6.4 Esybių ryšių diagrama 38
Rezultatai ir išvados **39**
Literatūros sąrašas **41**
Priedai **43**

Sutartinių terminų sąrašas

Stenoterminis - prisitaikęs gyventi, augti nedidelio aplinkos temperatūros svyravimo sąlygomis.

Termistorius - temperatūrai jautrus rezistorius, kurio varža kinta dėl jį veikiančios temperatūros pokyčio.

Anotacija

Šiame darbe apžvelgtas Langstroth‘o avilys ir jo struktūra, apžvelgti elektroninio monitoringo privalumai, lyginant su rankiniu tikrinimu, aprašyti stebimi parametrai. Taip pat apžvelgta vokiečių programuotojo Maximilian’o Michels’o sukonstruota monitoringo sistema. Sukurtas monitoringo sistemos prototipas, leidžiantis stebėti avilio temperatūrą.

Raktiniai žodžiai: Arduino plokštė, bičių avilių monitoringas, temperatūros matavimas, temperatūros stebėjimas, termistorius, Langstroth‘o avilys, eskizas, Steinhart-Hart‘o lygtis.

Summary

In this paper the author reviews the Langstroth hive and its structure. The qualities of electronic hive monitoring compared to manual inspection, and monitored parameters are also reviewed. Furthermore, a beehive monitoring system, designed by a german programmer Maximilian Michels is reviewed. Finally a prototype of a monitoring system to monitor beehive temperature was created.

Keywords: Arduino board, beehive monitoring, temperature measuring, temperature monitoring, thermistor, Langstroth hive, sketch, Steinhart-Hart equation.

Įvadas

Bičių avilių monitoringas yra idėja, siekianti net XX a. pradžią. Amerikiečių bitininkystės profesorius Gates‘as N. Burton‘as metus stebėjo bičių avilio temperatūros ir svorio pokyčius, kad geriau suprastų bičių žiemojimo ypatumus [Bur14]. Pagrindinė problema yra ta, jog, norint sužinoti, kokia yra avilio būsena ir svarbūs parametrai, norint išlaikyti sveiką avilį, tenka tiesiogiai tikrinti avilio vidų, sutrikdant jo veiklą. Tačiau, net toks būdas nėra labai informatyvus. Monitoringas sprendžia šitą bėdą - naudojant elektroninius daviklius galima nenutrūkstamai stebėti ir fiksuoti daugelį avilio parametrų, rinkti ir analizuoti informaciją realiu laiku, nesutrikdant avilio harmonijos. Tai yra ypač aktualu žiemą, kai avilio tikrinimas gali sukelti pavojų bičių gyvybėms.

Šio darbo tikslas yra sukonstruoti sistemą temperatūros matavimui, kurią sudaro programuojama elektroninė plokštė Arduino ir termistorius, ir nustatyti, ar tokia sistema gali būti pritaikoma aviliui. Šiam tikslui pasiekti darbe bus vykdomi tokie uždaviniai:

1. Aptarti Langstroth’o avilio struktūrą ir ypatumus.
2. Išsiaiškinti pagrindinius stebimus avilio būsenos parametrus.
3. Apžvelgtas vienas elektroninio bičių avilių monitoringo sistemos bandymas.
4. Sudaryti monitoringo sistemos principinę schemą.
5. Parašyti programinį kodą temperatūros davikliams, praktiškai pamatuoti temperatūrą ir daviklius palyginti.
6. Sukurti monitoringo sistemos prototipą.

Detali informacija apie Langstroth‘o avilio tipą pateikiama amerikiečio, entomologijos profesoriaus, Gene Kritsky knygoje „*The quest for the perfect hive: a history of innovation in bee culture“*. Joje apžvelgiama kaip amerikiečių bitininkas Lorens‘as Lorraine‘as Langstroth‘as sukūrė tarprėmį ir panaudojo jį konstruodamas naujo tipo avilį, kurį užpatentavo 1852 metais. Apie šio tipo avilio struktūros ypatumų aprašymą taip pat rašoma bitininko Howland Blackiston knygoje „*Beekeeping For Dummies. 2nd Edition“*.

Aptardamas esamus elektroninio bičių avilių stebėjimo tyrimus apžvelgiu Maximilian‘o Michels‘o studijoje „*A Beehive Monitoring System Incorporating Optical Flow as a Source of Information“* sukonstruotos avilio monitoringo sistemos dalis ir jų pritaikymą.

Daug informacijos apie patį monitoringą, jo istoriją ir pritaikymą galima rasti moksliniame W.G. Meikle, N. Holst straipsnyje „Application of continuous monitoring of honeybee colonies“.

1. Tradiciniai ir Langstroth‘o aviliai

1.1. Tradiciniai aviliai

Iki XIX a. aviliai dažniausiai būdavo natūralios erdvės, kuriose galėdavo gyventi bitės. Tradiciniai aviliai neturėdavo vidinės struktūros, todėl bičių sukurti koriai kryžiuodavosi ir įgaudavo naturalią chaotišką formą. Todėl medaus išėmimas paprastai reikšdavo avilių sunaikinimą. Šiais laikais tokie aviliai nėra plačiai naudojami (o daugelyje šalių net yra uždrausti), nes nėra galimybės patikrinti bičių ar korio būsenos nesunaikinant korio ar net visos bičių kolonijos [Cra79].

Bitininkystė iš esmės pasikeitė 1814 metais, kai ukrainietis Petro Prokopovičius išrado pirmąjį pasaulyje keičiamų rėmų sistema paremtą avilį. Tokie aviliai buvo sukurti, siekiant sukelti kuo mažiau žalos bitėms. Išimamų rėmų viduje bitės lipdo korį tolygia plokštuma, todėl juos galima išimti iš avilio kartu su visu rėmu, nepažeidžiant šalia esančių korių ir nesutrikdant bičių ramybės. Šios sistemos sukūrimas žymėjo komercinės bitininkystės pradžią [Sra02].

1.2. Langstroto avilys

Vienas dažniausiai naudojamų bičių avilių tipų yra Langstroto (angl. *Langstroth*) avilys, kuris buvo užpatentuotas amerikiečių bitininko Lorens‘o Lorraine‘o Langstroth‘o 1852 metais. Studijuodamas bičių elgseną Langstroth‘as pastebėjo, kad bitės lipdo korius tam tikru atstumu vienas nuo kito. Šis atstumas vadinamas *tarprėmiu*. Kai atstumas yra didesnis nei ~9.5 mm bitės lipdo perteklinį korį. Kai atstumas yra mažesnis nei ~6.35 mm bitės užpildo šį atstumą pikiu. Dėl šios priežasties daugelis bitininkų pasirenka ~8mm atstumo vidurkį.

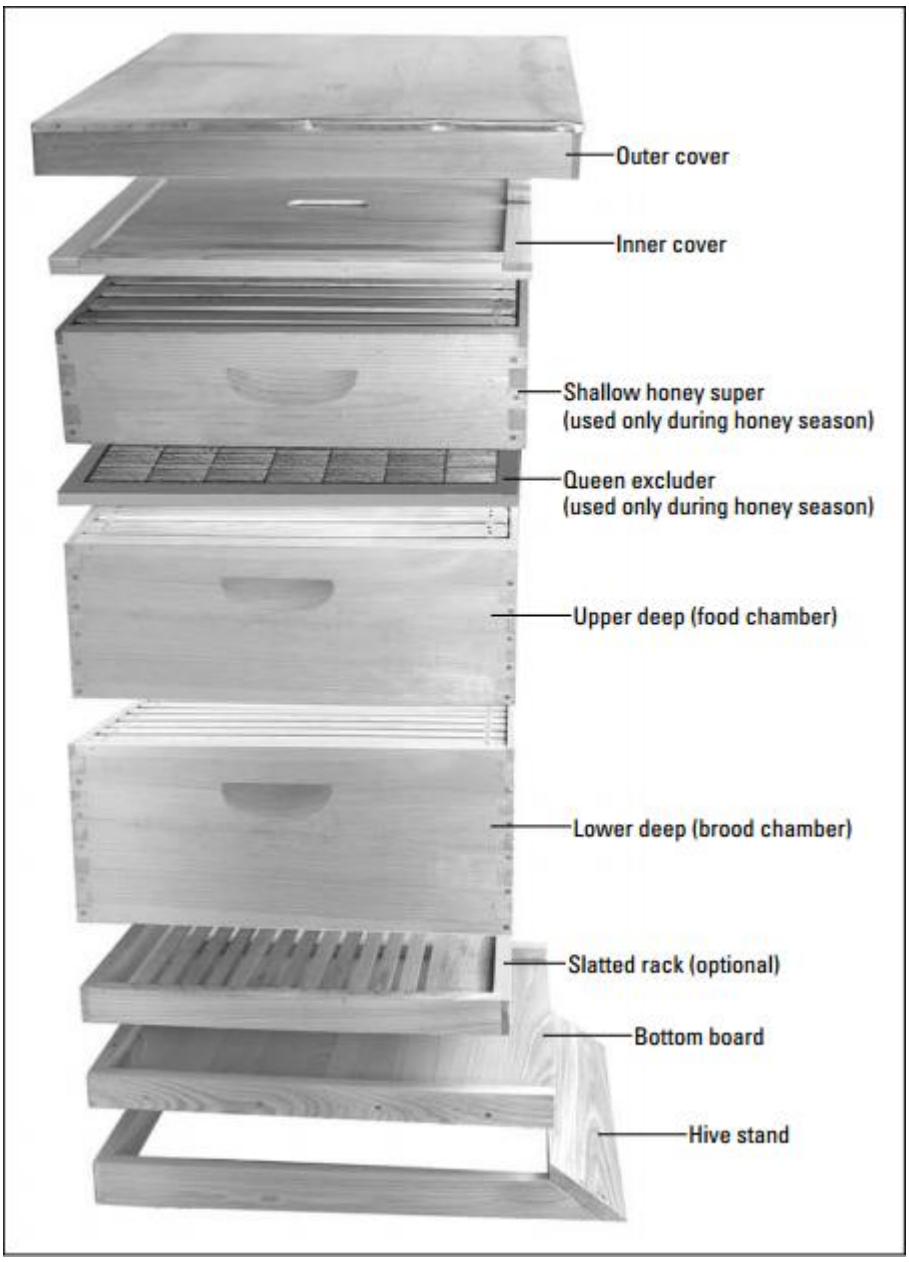
Langstroto avilio tipas yra paremtas vertikalia moduline konstrukcija, kuri suteikia galimybę lengvai pridėti arba nuimti avilio dalis. Avilys tapo populiarus dėl išimamų rėmų panaudojimo, perkeliamumo ir tarprėmio panaudojimo [Kri10].

1.3. Langstroth‘o avilio struktūra

Pagrindinės Langstroth‘o avilio dalys yra dėžės, dedamos viena ant kitos, ir jose išdėstomi rėmai koriams. Šiose avilio konstrukcijos dalyse bitės gyvena ir kaupia medų. Avilių dėžės būna kelių dydžių (arba gylių):

1. **Didelė (243 mm)** (angl. *deep body / deep super / brood chamber / hive body*). Ši dėžė paprastai naudojama bičių apgyvendinimui. Šilto klimato kraštuose dažniausiai užtenka vienos tokios dėžės, tačiau šalto klimato kraštuose bitininkai naudoja ir dvi tokias dėžes, kad bitės turėtų daugiau vietos gyvenimui bei galėtų sukaupti daugiau medaus. Šalto klimato vietovėse bitėms reikia daugiau medaus, kuris padeda išgyventi sunkiu laikotarpiu, pvz., žiemą. Pilna medaus šio tipo dėžė gali labai daug sverti, todėl dažnai yra naudojama kaip bičių šeimos lizdas (ang. *brood box*).
2. **Vidutinė (170 mm)** (angl. *medium / Western / Illinois Super)*. Tokio gylio dėžės dažniausiai naudojamos medaus kaupimui, nes pilnos sveria mažiau negu gilios dėžės. Bitininkai, norėdami išvengti rūpesčių dėl skirtingų išmatavimų rėmų vienam aviliui, kartais naudoja tik vidutines dėžes (vietoje dviejų gilių dėžių galima naudoti tris vidutines lizdui arba dvi vidutines vietoje vienos gilios medaus kaupimui). Tačiau, šios dėžės yra žemesnės nei gilios, jų reikia daugiau negu gilių dėžių, o tai padidina naudojamų rėmų skaičių, o tuo pačiu ir darbo laiką, reikalingą medaus išgavimui.
3. **Maža (150 mm)** (angl. *shallow*). Lengviausia dėžė iš minėtų. Dažniausiai naudojama medaus kaupimui, tačiau didesnis rėmų skaičius dar labiau padidina darbo laiko sanaudas. Yra ir 120 mm tokios dėžės variantas, kuris specialiai naudojamas tik medaus išgavimui. Dar vadinama meduve.

Visų tipų dėžės būna kelių standartinių pločių, matuojamų rėmais. Langstroth‘o aviliuose naudojama 8 - 10 rėmų. Lizdams formuoti yra patogiau naudoti 10 rėmų, o medui - 8 rėmus, nes pilnos medaus dėžės sveria pakankamai daug.



1 pav. Langstroth‘o avilio struktūra. Adaptuota iš [Bla09]

Langstroth‘o avilio struktūra detaliau [Bla09]:

1. **Stovas** (angl. *stand*) – pradinė avilio dalis, kuri pakelia jį nuo žemės. Gali turėti pakreiptą nusileidimo plokštę – prielaktį - bitėms. Nebūtina, tačiau dažniausiai naudojama dėl savo suteikiamų pliusų: apsaugo avilį nuo drėgmės ir galimo grybelio užsiveisimo, pašalina lengvą priėjimą įvairiems plėšrūnams, pagerina ventiliaciją.
2. **Avilio dugnas** (angl. *bottom board*) – ši dalis dedasi ant stovo. Gali būti vientisa arba sietinė. Sietinė apatinė lenta pagerina oro cirkuliaciją ir neleidžia susidaryti kondensacijai. Su tokia

lenta galima naudoti ir erkėgaudį (angl. *sticky board*). Bitininkai naudoja ją bičių erkėtumo kontrolei.

3. **Įėjimo reduktorius / lakos kamštis** (angl. *entrance reducer*) – ši detalė leidžia bitininkui reguliuoti avilio eismą, temperatūrą ir ventiliaciją. Dedama tarp apatinės lentos ir pirmosios spiečiaus dėžės. Neturint šios dalies galima naudoti lapus, žolę arba švarias audeklo atraižas.
4. **Lizdas** (angl. *brood box / brood chamber / hive body*) – didžiausia avilio dėžė. Galima naudoti dvi tokias dėžes vieną ant kitos – apatinę bitės naudoja dauginimuisi, o viršutinėje laiko medų ir žiedadulkes savo reikmėms.
5. **Bičių tvorelė** (angl. *Queen excluder*). Neprivaloma dalis, kuri yra dedama tarp spiečiaus ir medaus dėžių. Naudojama tam, kad bičių motinėlė nepatektų į medaus dėžę ir nepradėtų meduvėse dėti kiaušinėlių. Kai kurie bitininkai neigiamai vertina šios dalies naudojimą ir vadina medaus sandarikliu/pašalintoju (angl. *honey excluder*). Problemos kyla, kai bitėms sunku ar nepatinka pereiti per šią dalį (pasak tam tikrų teorijų, bitės darbininkės nelinkusios judėti ten, kur negali motinėlė), todėl medus gali būti pradedamas kaupti lizde, o tai gali sąlygoti *spietimąsį* (motinėlės išėjimas iš avilio kartu su didele grupe bičių darbininkių ir naujos bičių kolonijos sukūrimas).
6. **Meduvė** (angl. *super / honey super*) – šios dėžės rėmuose bitės kaupia medų, kurį renka bitininkai. Pertekliaus laikotarpiu galima sukrauti daug tokių dėžių.
7. **Vidinis dangtis / vidinė danga** – dedamas tarp viršutinės medaus dėžės ir išorinio dangčio. Turi skylę viduryje, kuri gerina ventiliaciją ir mažina kondensaciją avilyje, taip pat išpjovą viename šone. Ši dalis yra viena svarbesnių dėl kelių priežasčių. Pirma – bitės labai dažnai užsandarina tarpus propoliu. Jeigu nebūtų vidinio dangčio, bitės užsandarintų išorinį dangtį, kurį yra labai sunku nuimti (ypač jeigu jo šonai apglėba avilį),. Ne be reikalo propolis yra vadinamis bičių avilio cementu. Antra – skylė dangčio viduryje suteikia galimybę lengvai pamaitinti bites. Šoninė skylė taip pat turi privalumų. Pirmiausia – ji tarnauja kaip dar vienas įėjimas bitėms. Kadangi bitės kaupia medų aukštesniosiose avilio dalyse, joms gali būti sunku nešti medų visą kelią nuo apačios, ypač jeigu yra daug medaus dėžių. Antra – šoninė skylė gali būti naudojama kaip orlaidė. Jos pagalba sukuriamas oro srautas, kuriuo yra vėdinamas avilys, sumažinama drėgmė ir į avilį patenka šviežias oras. Tai gali būti labai pravartu žiemą, kuomet dėl bičių kvėpavimo avilyje kondensuojasi oras ir pradeda kauptis drėgmė.
8. **Stogas** (angl. *Outer cover)* – viršutinė avilio dalis, dažniausiai gaminama iš medžio ir padengiama skarda. Didžiausias šios dalies pliusas – bitininkas lengvai gali prieiti prie avilio

vidaus ir patikrinti bičių šeimą arba pasiekti korius. Ši dalis gaminama didesnė negu kitos tam, kad norėdamas bitininkas galėtų ją judinti. Išorinio dangčio judinimas suteikia galimybę uždaryti vidinio dangčio šoninę skylę, taip apribojant orlaidę bei neleidžiant bitėms naudoti skylės įėjimui. Taip pat ši dalis apsaugo avilį nuo lietaus bei vėjo.

2. Bičių avilių monitoringas

2.1. Bičių avilių tikrinimas

Bičių avilių tikrinimas yra svarbus procesas, užtikrinantis avilių kokybę bei bičių sveikatą. Tikrindamas avilį bitininkas apžvelgia bites ir avilio rėmų užimtumą, taip pat patikrina, ar nėra užsiveisusių kenkėjų. Apžiūrėjęs avilį, bitininkas apgalvoja tolimesnių veiksmų eigą, pvz., bičių pamaitinimą, šeimos padalijimą ar naujos motinėlės įdavimą. Norint užtikrinti gerą avilio būklę, būtina dažnai jį tikrinti, tačiau tai yra sunkus ir daug laiko užimantis darbas, kas mažina galimų patikrinti avilių kiekį. Taip pat, jeigu bitininkas prižiūri daug bitynų, tai dar labiau pasunkina tikrinimo užduotį, kadangi bityne gali būti ribotas avilių skaičius, priklausantis nuo to, kokie nektaro bei žiedadulkių šaltiniai yra bityno apylinkėse.

Patyręs bitininkas, stebėdamas avilį iš išorės, gali nuspręsti, ar verta tikrinti jo vidų pagal tai, kaip elgiasi bitės, kokį garsą jos skleidžia, kiek ir kokių bičių atskrenda ir išskrenda, kokį kvapą skleidžia avilys. Jeigu avilys skleidžia ramų garsą, gerą kvapą ir jo stovas švarus, greičiausiai viskas yra gerai ir jo tikrinti nereikia. Tačiau vertėtų avilį patikrinti, jeigu mažėja skraidančių bičių kiekis, aplink avilį guli negyvos bitės, iš avilio sklinda keistas kvapas, aplink avilį renkasi kenkėjai.

Avilio tikrinimas bitėms sukelia stresą, kuris kenkia bičių sveikatai ir sumažina produktyvumą. Kiekvieną kartą tikrinant avilį gali būti nužudomos bitės, tarp jų ir motinėlė, atšaldomas lizdas, sulaužomos propolio plombos, taip pat, pažeidus meduvių korius, kenkėjai gali užuosti medaus kvapą ir bandyti įsilaužti į avilį.

2.2. Elektroninis bičių avilių monitoringas

Pagrindinis bičių avilių monitoringo tikslas yra automatizuotas gyvybiškai svarbios informacijos apie bičių šeimą rinkimas pasinaudojant elektroniniais jutikliais.

Elektroninis bičių avilių monitoringas gali būti išskirtas į tris dalis [ZSM12]:

1. Informacijos rinkimą.
2. Informacijos analizę.
3. Informacijos pritaikymą.

Monitoringas yra geras būdas bitininkams suteikti pastovią informaciją apie bičių šeimos būklę. Šiomis dienomis gali būti stebimi daugelis parametrų, pvz., temperatūra, oro drėgmė, garsas, svoris, bičių judėjimas. Automatizuoto monitoringo sistema gali kaupti duomenis, kurie gali būti analizuojami ir naudojami išvadoms padaryti. Šiame skyriuje bus apžvelgti stebimi parametrai, apžvelgtos ir palygintos technologijos, naudojamos jų matavimui.

2.3. Stebimi parametrai

Svoris. Bičių avilio svoris yra vienas pagrindinių dydžių, nes jį sudaro bičių populiacija, medus ir žiedadulkės. Svorio matavimas yra geras būdas sužinoti informaciją apie bičių avilio produktyvumą, pvz., medaus ir/arba bičių kiekį, kiek medaus bitės atneša per medunešį arba kiek medaus suvartoja žiemodamos (šiuo atveju matuojant svorį galima žinoti, kiek medaus atsargų liko bitėms ir kada jas reikia pamaitinti cukraus sirupu). Taip pat svorio matavimas yra gerai tuo, kad nereikia atidarinėti avilio ir tyrinėti kiekvieno rėmo, taip nesukeliant streso sau ir bitėms [Eva15, MH15].

Temperatūra. Bičių jaunikliai yra stenoterminiai, todėl tinkamam jų vystymuisi yra būtinas tikslus temperatūros reguliavimas. Lizde bitės palaiko temperatūra tarp 33°C ir 36°C. Bitės gali efektyviai reguliuoti temperatūrą avilio viduje, todėl jos gali išgyventi daugelyje klimatų. Esant žemai temperatūrai bitės suformuoja žiemos kamuolį ir generuoja šilumą sutraukdamos ir atpalaiduodamos savo „pilvelio raumenis“ tokiu būdu generuodamos šilumą. Vasarą bitės ventiliuoja lizdą „vėduodamos“ karštą orą iš avilio; papildomai gali būti panaudojami vandens lašeliai, paskleisti po korį, vėdinimas sukelią jų garavimą, ko rezultatas yra aktyvus vėdinimas [FSp58].

Temperatūra – matmuo, kurį galima nesunkiai ir ekonomiškai stebėti. Duomenys apie temperatūrą gali suteikti informacijos apie su kolonija susijusias būsenas, pvz., padidėjusį maisto suvartojimą, perų auginimo pradžią, priešspietiminę būseną ar net kolonijos mirtį. [Eva15, MH15]

Drėgmė. Dėl daugelio priežasčių bičių šeimą įtakojantis parametras. Kelios iš jų:

- Optimali drėgmės riba spiečiaus veisimui yra tarp 90% ir 95%, tuo tarpu žemesnėje nei 50% drėgmėje kiaušinėliai neišsirita [Dou76].
- Didesnė drėgmė įtakoja mažesnį erkių veisimąsi peruose [KV97].

- Santykinei drėgmei pakilus nuo 67% iki 87% kalkinių perų (sukeltų askosferozės - grybinės ligos, nuo kurios perai sukalkėja) skaičius padidėjo 8% [FRR+96].

Anglies dioksidas. Šis parametras įtakoja bičių vėdinimo elgesį – padidėjus anglies dvideginio koncentracijai, padidėja bičių, vėdinančių avilį, skaičius. Žemas anglies dvideginio lygio palaikymas yra nuolatinis ir didesnėse bičių šeimose anglies dvideginio lygis yra pastovesnis dėl beveik nepertraukiamo vėdinimo [See74].

3. Bičių avilio monitoringo sistemos apžvalga

Maximilian Michels savo darbe „Bičių avilio monitoringo sistema įtraukiant optinį srautą kaip informacijos šaltinį“ [Mic11] sukonstravo sistemą bičių avilio monitoringui. Darbo tikslas buvo sukurti automatizuoto informacijos rinkimo sistemą, kuri pagelbėtų tyrinėjant netikėtą bičių kolonijų žlugimą (angl. *Colony Collapse Disorder (CCD)).* Sistema stebėjo keturis parametrus: avilio temperatūrą, svorį, aplinkos temperatūrą bei optinį srautą (angl. *Optical flow).* Stebėjimas vyko kontroliuojamoje aplinkoje. Visa įranga buvo prijungta prie kompiuterio A, kuris nuskaitė jutiklių duomenis bei apdorojo vaizdinę medžiagą. Ta informacija buvo perduodama kompiuteriui B, kuris skirtas informacijos kaupimui bei priėmimui per saityno serverį (angl. *Web server).* Jutikliai buvo nuskaitomi kas minutę. Jų duomenys būdavo kaupiami buferyje, nusiunčiami į kompiuterį B ir buferis būdavo išvalomas.

Svorio monitoringui autorius naudojo „PCE-PCS 30“ svarstykles (angl. *Counting scale).* Šios svarstyklės turi taros svorio (angl. *Tare weight)* funkciją, RS-232 standarto nuoseklųjį prievadą (angl. *Serial port)* duomenų perdavimui, kuris gali siųsti duomenis jiems pasikeitus arba laiko intervalais, jų svėrimo riba yra iki 30 kg su 0.5 kg paklaida, gali veikti 0°C - 40°C temperatūrų ribose. Kadangi autorius atliko stebėjimą kontroliuojamoje aplinkoje, šios svarstyklės buvo tinkamos, tačiau lauko aplinkoje reikėtų geresnio varianto, dėl šių priežasčių:

- Per maža svėrimo galia – kadangi sveriamas ne tik pats avilys, bet ir visa, kas yra jame, gali susidaryti didelis bendras svoris. Apytiksliai didelis lizdas gali sverti iki 36 kg, didelė meduvė tarp 32 ir 40 kg, vidutinė meduvė – tarp 18 ir 23 kg[1]. Bendras svoris gaunasi tarp 86 ir 100 kg, todėl svorio monitoringui šios svarstyklės nebetinka.
- Temperatūros riba – kadangi bitynas gali gyvuoti ne tik šiltuoju metu, bet ir šaltuoju, šios svarstyklės neigiamoje temperatūroje neveiks.
- Duomenų perdavimas – šiuo atveju duomenų perdavimui yra naudojami du kompiuteriai. Duomenų perdavimas vyksta laidais. Nors laidinis duomenų perdavimas yra pranašesnis už bevielį, pastarasis laimi patogumu. Stebint vieną avilį galima pasikliauti ir laidais, tačiau bitynus dažniausiai sudaro daugiau negu vienas avilys, o tai reiškia laidų tinklą.

[1] Autoriaus pastaba: avilys nebūtinai turi susidėti iš šių dalių.

Avilio temperatūros monitoringui buvo naudojamas „DS1820“ skaitmeninis termometras. Šis termometras turi vieno laido sąsają (angl. *One-wire interface).* Tai reiškia, kad duomenų siuntimui ir perdavimui užtenka tik vieno prievado (angl. *Port).* Taip pat, termometras gali būti maitinamas ta pačia sąsaja, todėl nereikia išorinio maitinimo šaltinio. Temperatūros nuskaitymui buvo naudojama atviro kodo programinė įranga »digitemp«. Ši programinė įranga palaiko ne tik minėtąjį skaitmeninį termometrą, tačiau ir daugelį kitų (reikėtų pabrėžti, kad visi jie sukurti tos pačios kompanijos - Maxim Integrated). Temperatūros nuskaitymui gali būti naudojami ir keli tokie termometrai, tereikia juos sujungti šakotuvu (angl. *Hub)*. Šie jutikliai turi unikalų 64-bitų serijinį numerį, skirtą identifikavimui. Jie gali veikti -55°C ir 125°C laipsnių ribose, kurios yra pakankamos daugelio pasaulio vietų klimate. Taip pat jie veikia 0.5°C laipsnio tikslumu -10°C ir 85°C laipsnių ribose.

Aplinkos oro monitoringui buvo naudojama „WS-2310“ meteorologinė stotis su „Heavy Weather“ programine įranga. Ši stotis turi jutiklius vidinei temperatūrai ir drėgmei matuoti, taip pat stotelė lauko parametrams (temperatūrai, drėgmei, vėjo greičiui ir krypčiai bei lietui) matuoti. Visi jutikliai yra prijungti prie bazinės stotelės. Bazinė stotelė gali būti varoma baterijomis arba AC/DC adapteriu. Lauko stotelė gali perduoti duomenis bazinei stotelei arba kabeliu, arba bevieliu 433 MHz dažniu. Programinė įranga kas minutę nuskaito jutiklius ir surašo duomenis į tekstinį failą. Vartotojo sąsaja suteikia galimybę matyti momentinius ir istorinius duomenis.

Nors ir autoriaus naudotos svarstyklės netiktų naudojimui lauko sąlygose, tačiau tiktų naudotas termometras. Taip pat, Arduino programėlėje galima panaudoti papildomą biblioteką, kurios pagalba galima iškart išgauti temperatūros reikšmę, neatliekant papildomų skaičiavimų (tokius skaičiavimus pateiksiu rašydamas programėlę temperatūros stebėjimui termistoriumi).

4. Arduino plokštė ir principinė schema

4.1. Kas yra Arduino Uno?

Arduino Uno yra atviro naudojimo (angl. *Open-source)* programuojama elektroninė plokštė, skirta elektroninių projektų kūrimui. Arduino Uno susideda iš programuojamo Atmel mikrovaldiklio (angl. *Microcontroller)* ir integruotos kūrimo aplinkos (angl. *Integrated Development Environment* arba *IDE)*, kuri suteikia galimybę lengvai rašyti kodą ir įkeltį jį į plokštę.

4.2. Kodėl Arduino Uno?

Ši plokštė yra populiari, nes su ja paprasta pradėti dirbti, nebereikia nieko lituoti. Net ir neturėdamas programavimo žinių žmogus gali nesunkiai išmokti programuoti ir parašyti programą. Taip pat ši platforma turi ir kitų pranašumų:

- Arduino plokštės yra gana pigios palyginus su kitomis plokštėmis.
- Arduino programinė įranga veikia Windows, Mac OSX ir Linux operacinėse sistemose.
- Arduino yra atviro naudojimo. Tai reiškia, kad programinė bei techninė įranga yra lengvai prieinama ir modifikuojama bei plečiama.
- Lengvai naudojama – jungiasi prie kompiuterio USB kabeliu.

Taip pat Arduino gali būti praplečiamas skydeliais (angl. *Shield)* – modulinėmis plokštėmis, kurios jungiasi prie Arduino ir suteikia galimybę, pvz., prijungti plokštę prie interneto, bevielio tinklo. Skydeliams dažniausiai yra suteikiamos bibliotekos ir pavyzdiniai eskizai, todėl naudotis jais nėra sudėtinga, tiesiog reikia prijungti prie plokštės ir įkelti į ją kodą. Dauguma skydelių gali būti kraunami vienas ant kito, turint omenyje, kad jie yra suderinami su plokšte ir nenaudoja sutampančių kištukų.

4.3. Iš ko susideda Arduino programėlė?

Programos, parašytos Arduino programine įranga, vadinamos eskizais (angl. *Sketch)* ir naudoja *.ino* failo plėtinį (angl. *File extension).* Kiekvieno eskizo kodas susideda iš dviejų pagrindinių tuščio (angl. *Void)* tipo funkcijų: *setup()* ir *loop()*. Pirmoji funkcija yra paleidžiama vieną kartą įjungus

plokštę. Šitoje funkcijoje yra nustatomi komunikacijos su kompiuteriu parametrai. Po to kartojasi antroji funkcija. Šioje funkcijoje yra rašomas kodas, kuriuo norima kontroliuoti plokštę.

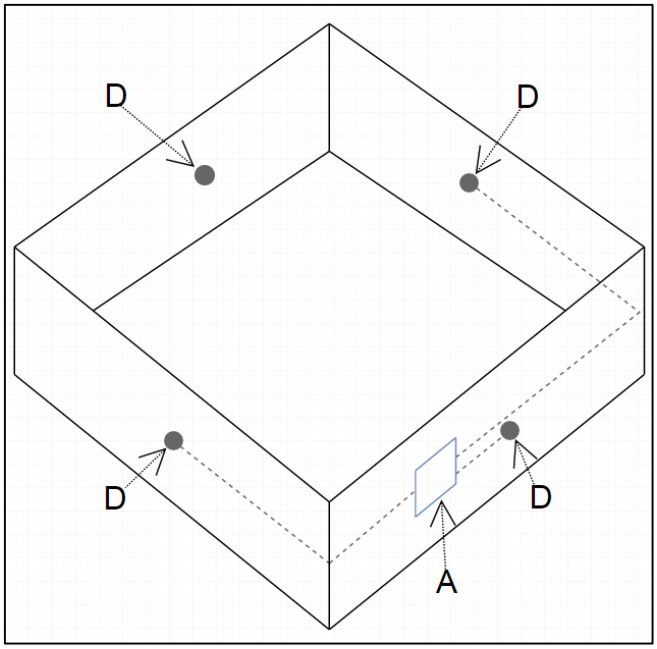
```
01 void setup() {
02 //inicializacijos žingsniai
03 }
04
05 void loop() {
06 //darbinis kodas
07 }
```

2 pav. „Plikos“ programos pavyzdys

2 pav. pateiktas minimalaus kodo pavyzdys, leidžiantis susikompiliuoti eskizą.

4.4. Arduino plokštės ir daviklių pajungimas prie modulinio avilio

Vienas iš galimų būdų plokštės ir daviklių pritvirtinimui prie avilio yra pavaizduotas 3 pav. Plokštė (**A**) yra pritvirtinama prie dėžės šono ir davikliai (**D**) yra kabinami dėžės viduje ant sienų. Po to užtenka prijungti maitinimą.



3 pav. Modulinio avilio temperatūros stebėjimo sistemos principinė schema[2]

[2] Schemos piešimui buvo naudota nemokama internetinė diagramų piešimo programinė įranga „Draw.io“. Prieiga per internetą: https://www.draw.io/

Privalumai taip prijungiant daviklius:

- Nukeliant dėžę nereikia perjungti daviklių.
- Lieka tik užtikrinti elektros šaltinį.
- Sistema tampa itin mobili.

4.5. Temperatūros matavimas su Arduino Uno

Toliau mėginsime praktiškai pamatuoti temperatūrą. Temperatūros matavimas yra gana paprastas žingsnis išmėginant anksčiau aprašytos idėjos įrodymą. Tiek Arduino plokštė, tiek temperatūriniai davikliai yra pakankamai pigūs ir lengvai montuojami, todėl jie buvo pasirinkti šiam etapui.

5. Praktinis temperatūros matavimas

5.1. Pasiruošimas darbui

Iš pradžių reikia pasiruošti darbui. Šis pasiruošimas susideda iš dviejų dalių:

1. Reikiamos įrangos surinkimas.
2. Darbo aplinkos pasiruošimas.

Temperatūros matavimo įgyvendinimui reikės kelių dalykų:

- Arduino plokštės (šiuo atveju naudosiu Arduino Uno modelį)
- Termistoriaus (šiuo atveju naudosiu IM120628010 modelį)
- Skaitmeninio termometro (šiuo atveju naudosiu DS18B20 modelį)
- Maketavimo plokštės (angl. *Breadboard)*
- Rezistoriaus arba varžos
- Jungiamųjų laidų

Reikiamos įrangos surinkimas prasideda plokštės prijungimu prie kompiuteriu. Taip pat reikia parsisiųsti ir sukonfiguruoti programinę įrangą. Plokštė yra prijungiama USB A/B kabeliu. Kaip minėta anksčiau, programėlės Arduino plokštei yra rašomos su Arduino programine įranga. Parsisiuntus ir įrašius ją į kompiuterį, reikia pasirinkti naudojamos plokštės modelį (tai galima padaryti mygtukų juostoje (angl. *Toolbar)* spaudžiant **Tools** > **Board**) ir nuoseklųjį prievadą, kurį naudoja prijungta Arduino plokštė (jei plokštė gerai prijungta, turėtų matytis pasirinkimas **Tools** > **Port** meniu).

Sekantis žingsnis yra visų dalių sujungimas. Prieš matuojant temperatūrą reikia žinoti kas yra termistorius ir kaip jis veikia.

5.2. Termistorius

Termistorius yra „terminis rezistorius“, t.y. rezistorius, kurio varža keičiasi keičiantis temperatūrai. Yra dviejų tipų termistoriai: teigiamo temperatūros koeficiento (angl. *PTC – positive temperature coefficient),* kurių varža didėja, kylant temperatūrai, arba neigiamo temperatūros koeficiento (angl.

NTC – negative temperature coefficient), kurių varža mažėja, kylant temperatūrai. Temperatūros matavimui dažniausiai naudojami pastarieji [SM97]. Termistoriai yra populiarūs dėl to, kad jais lengva naudotis, jie greitai reaguoja į temperatūros pokyčius ir yra pigūs [Cig06].

5.3. Analoginis-skaitmeninis keitiklis

Prieš apskaičiuojant temperatūrą reikia apskaičiuoti termistoriaus varžą. Arduino plokštė tiesiogiai matuoti varžos negali, tačiau gali matuoti įtampą su įtaisytu ASK (analoginis-skaitmeninis keitiklis; angl. *Analog-to-digital converter).* Arduino turi šešis ASK kištukus, kurie paverčia išėjimo įtampą į skaitmeninę reikšmę. Šie kanalai yra 10 bitų rezoliucijos, o tai reiškia, kad kiekvienas kanalas gali grąžinti $2^{\wedge 10}$ arba 1024 skaitmeninių reikšmių. Arduino numatytoji darbinė įtampa yra 5V, todėl ASK priskiria skaitmenines reikšmes [0, 1023] atitinkamoms įtampoms [0V, 5V]. Arduino ASK sieja įtampą su skaitmenine reikšme pagal sekančią formulę:

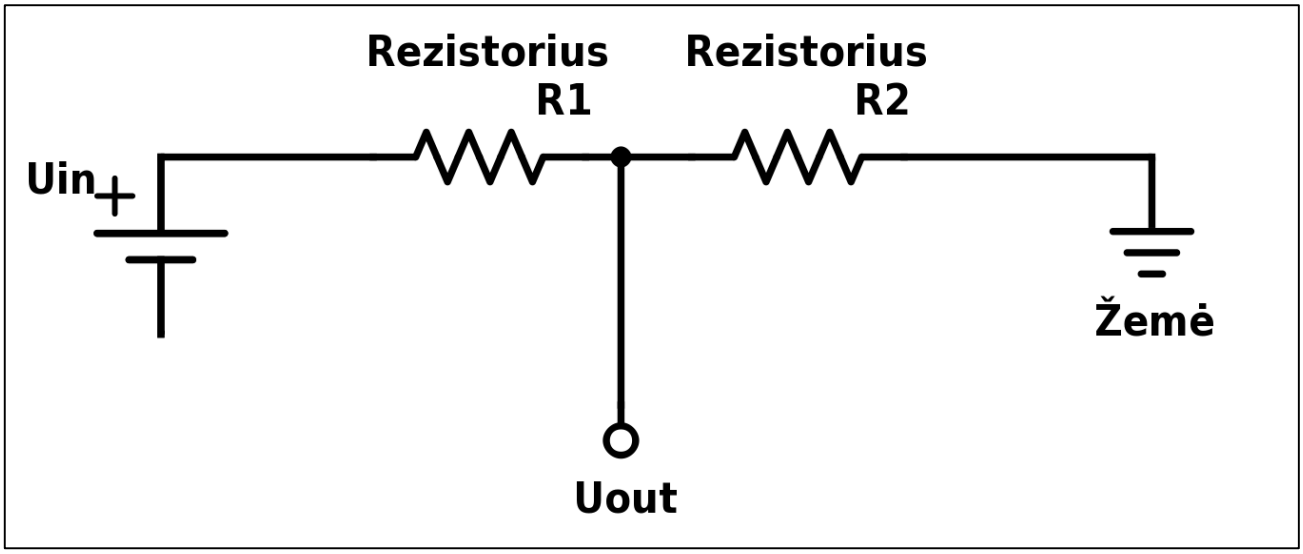
$$ASK\ reikšmė = \frac{ASK\ rezoliucija}{Darbinė\ įtampa} \times\ Nuskaityta\ įtampa$$

Pagal naudojamos Arduino plokštės rezoliuciją ir darbinę įtampą formulę galima supaprastinti:

$$ASK\ reikšmė = \frac{1023}{5V} \times\ Nuskaityta\ įtampa$$

5.4. Įtampos daliklis

Paprastai paversti kintančią termistoriaus varžą į įtampą galima panaudojant termistorių įtampos daliklyje (angl. *Voltage divider).* Įtampos daliklis yra elektros grandinė, paverčianti didelę įtampa mažesne (4 pav.).



4 pav. Įtampos daliklio grandinė

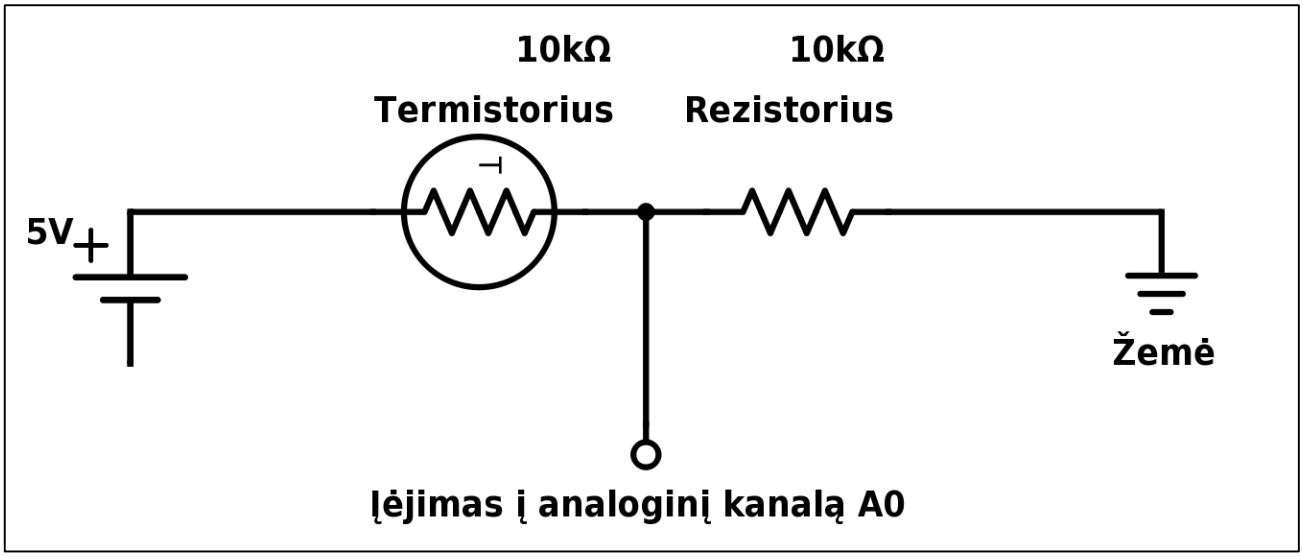
Įtampos daliklio lygties pagalba galime rasti išėjimo įtampą (U_{out}). Svarbiausia žinoti tris lygties dydžius: U_{in} ir rezistorių R1 ir R2 varžas.

$$U_{out} = \frac{R2}{R1 + R2} \times U_{in}$$

Pastaba: kai rezistorių varžos lygios, tai išėjimo įtampa lygi pusei įėjimo įtampos.

5.5. Termistoriaus grandinė

Termistoriaus grandinė yra tokia pati, kaip įtampos daliklio, tik vietoj vieno rezistoriaus yra jungiamas pats termistorius. Kadangi termistoriaus varža keičiasi keičiantis temperatūrai, keičiasi ir išėjimo įtampa (šiuo atveju tai yra analoginis kanalas A0) (5 pav.).



5 pav. Termistoriaus grandinė

5.6. Eskizas

Pasiruošę darbui galime pradėti rašyti eskizą temperatūros matavimui.

```
01 void setup() {
02     Serial.begin(9600);
03 }
04
05 void loop() {
06     float analogVal = analogRead(0);
07     Serial.print("Analog reading: ");
08     Serial.print(analogVal);
09
10     float resistance = 10000.0f * (1023.0f / analogVal - 1.0f);
11     Serial.print(" Resistance: ");
12     Serial.print(resistance);
13
14     delay(1000);
15 }
```

6 pav. Pradinio eskizo kodas

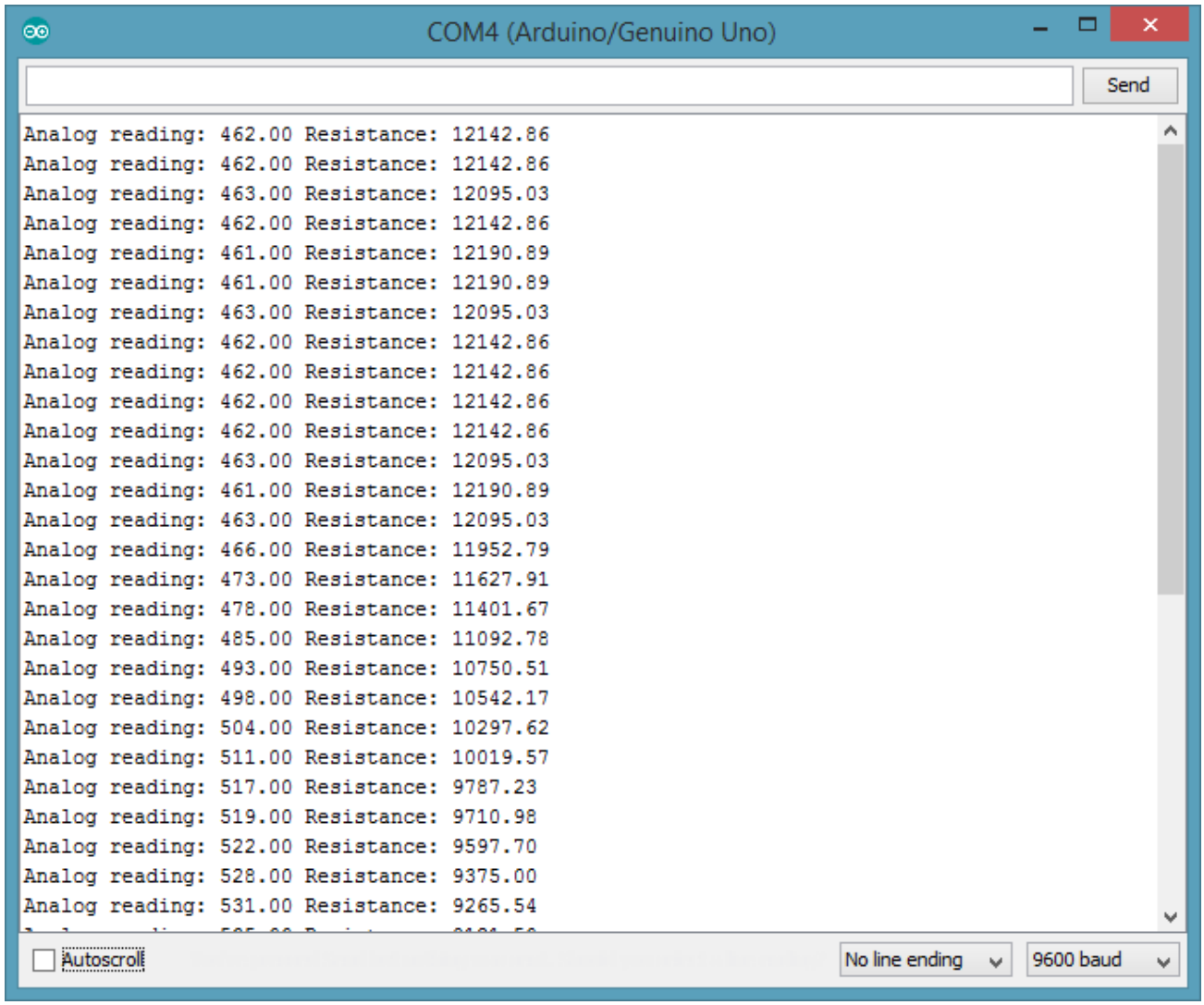
setup() funkcijoje nustatome duomenų perdavimo tarp plokštės ir kompiuterio greitį – 9600 bitų per sekundę. 9600 bitų yra dažniausiai naudojamas greitis.

loop() funkcijoje aprašome sveikojo skaičiaus tipo kintamąjį, kuriam priskiriame skaitmeninę reikšmę, nuskaitytą iš analoginio kanalo A0. Toliau išspausdiname tekstą bei kintamojo reikšmę, konvertuojame reikšmę į varžą pagal formulę:

$$R_t = R \times \left(\frac{V_s}{V_o} - 1\right)$$

čia **R** – rezistoriaus varža, **Vs** – įėjimo įtampa, **Vo** – išėjimo įtampa. Išspausdiname tekstą ir skaitmeninės reikšmės varžos atitikmenį.

Įkėlus kodą į plokštę, įjungiame srauto monitorių (angl. *Serial Monitor)*, paspaudę **Tools > Serial Monitor** arba **Ctrl + Shift + M** mygtukų kombinaciją. Srauto monitorius suteikia galimybę siųsti žinutes Arduino plokštei, taip pat gauti. Po kurio laiko 7 pav. matome spausdinamą tekstą ir reikšmes.



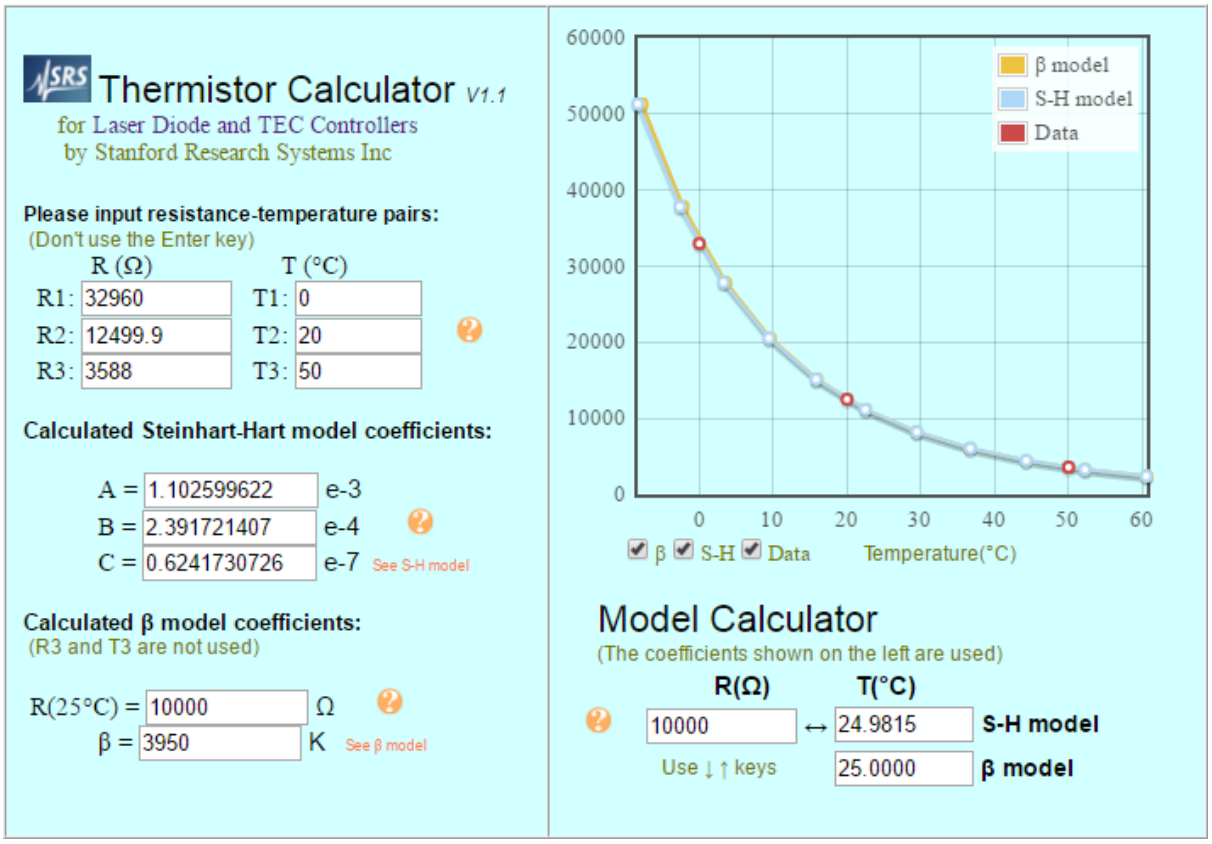
7 pav. Pradinio eskizo kodo rezultatas srauto monitoriuje

Maždaug ties viduriu galime pamatyti reikšmės padidėjimą. Tuo metu termistorius buvo suspaustas pirštais.

Naudojantis duomenų lapu (angl. *Datasheet)* galima nustatyti apytikslę varžų temperatūrą, jeigu duomenų lapas turi temperatūrų ir jų atitinkamų varžų lentelę (žr. 1 priedą), tačiau tai gali būti nepatogu, taip pat gali netenkinti lentelės tikslumas (šiame darbe naudojamos termistoriaus lentelėje surašytos reikšmės sveikojo skaičiaus tikslumu). Todėl reikia parašyti funkciją, paverčiančią skaitmeninę reikšmę į norimo matavimo vieneto temperatūrą. Dažniausiai naudojama lygtis termistoriaus temperatūrų ir varžų priklausomybei apskaičiuoti yra **Steinhart-Hart'o** lygtis:

$$T = \frac{1}{A + B \times \ln(R) + C \times \ln(R)^3}$$

čia **T** – temperatūra kelvinais, **R** – varža, o **A, B** ir **C –** Steinhart-Hart'o koeficientai termistoriui. Šios konstantos gali būti pateiktos duomenų lape, tačiau šiuo atveju mums reikia jas apskaičiuoti patiems arba naudojantis internetiniais kalkuliatoriais (žr. 2 priedą), palengvinančiais darbą. Reikia pasirinkti trijų temperatūrų ir jų atitinkamų varžų reikšmes, o kalkuliatorius atliks skaičiavimus (8 pav.).



8 pav. Termistoriaus kalkuliatorius. Prieiga per internetą:
http://www.thinksrs.com/downloads/programs/Therm%20Calc/NTCCalibrator/NTCcalculator.htm

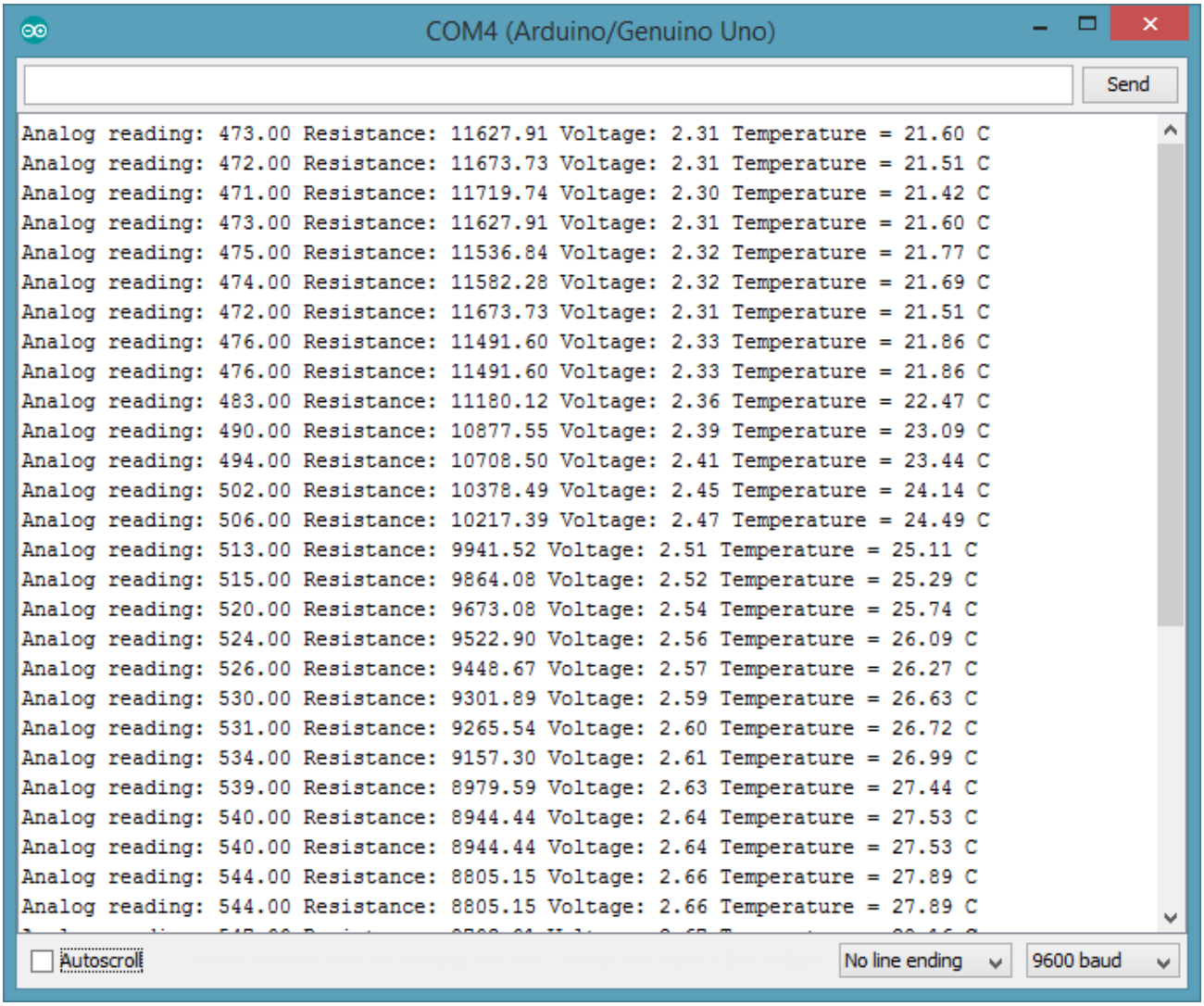
8 pav. matome, jog pasirinkus varžų {R1, R2, R3} reikšmes {32960, 12499.9, 3588} ir atitinkamų temperatūrų {T1, T2, T3} reikšmes {0, 20, 50} gauname A, B ir C koeficientų reikšmes, kurias galime naudoti savo eskize (9 pav.).

```
01 double convertToTemperature(float resistance) {
02     double temperature;
03     float a = 0.001102599622, b = 0.0002391721407, c = 0006241730726;
04
05     temperature = log(resistance);
06     temperature = 1 / (a + b * temperature + c * temperature * temperature
* temperature);
07     temperature = temperature - 273.15;
08     return temperature;
09 }
10
11 void setup() {
12     Serial.begin(9600);
13 }
14
15 void loop() {
16     float analogVal = analogRead(0);
17     Serial.print("Analog reading: ");
18     Serial.print(analogVal);
19
20     float resistance = 10000.0f * (1023.0f / analogVal - 1.0f);
21     Serial.print(" Resistance: ");
22     Serial.print(resistance);
23
24     float voltage;
25     voltage = analogVal * 5.0f / 1023.0f;
26     Serial.print(" Voltage: ");
27     Serial.print(voltage);
28
29     double temp = convertToTemperature( resistance);
30     Serial.print(" Temperature = ");
31     Serial.print(temp);
32     Serial.println(" C");
33
34     delay(1000);
35 }
```

9 pav. Galutinio eskizo kodas

loop() funkcijoje papildomai atspausdiname įtampą ir temperatūrą Celsijaus laipsniais. Temperatūrą gauname panaudodami **convertToTemperature** funkciją, kuri priima vieną parametrą – varžą. Ši funkcija naudoja Steinhart-Hart‘o lygtį varžos į temperatūrą konvertavimui, taip pat paverčia temperatūrą iš Kelvino į Celsijaus skalę.

Įkėlus papildytą kodą į plokštę ir kurį laiką palaukus, matome spausdinamus papildomus parametrus ir jų reikšmes:



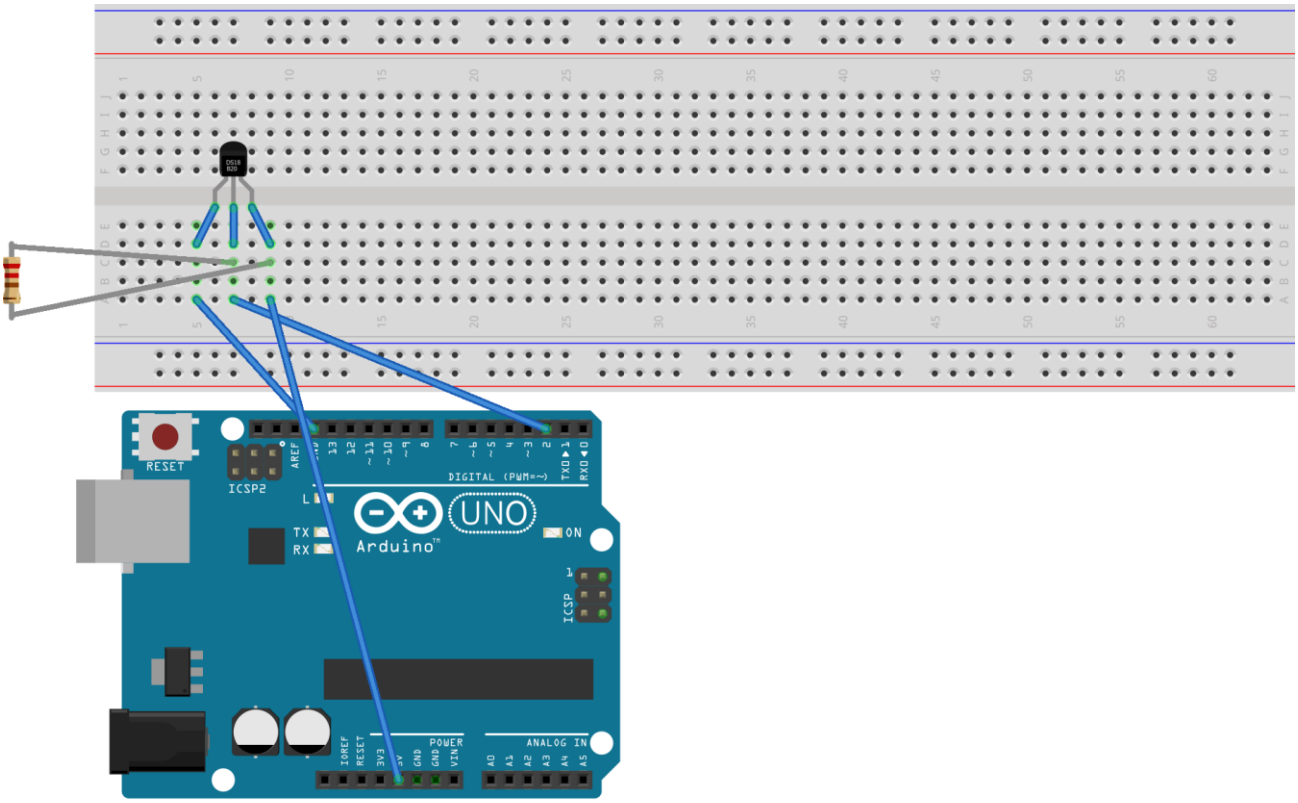
10 pav. Galutinio eskizo rezultatas srauto monitoriuje

Iš pradžių termistorius matavo kambario temperatūrą. Vėliau jis buvo paimtas į ranką, todėl jo rodoma temperatūra pradėjo kilti (10 paveikslas).

5.7 Temperatūros matavimas skaitmeniniu DS18B20 termometru

Šiame skyrelyje bus aprašytas „DS18B20“ skaitmeninis termometras, minėtas apžvelgtame Maximilian‘o Michels‘o darbe apie sistemą bičių avilio monitoringui. Mano darbe šis termometras bus sujungtas su Arduino, išbandytas matuojant temperatūrą ir palygintas su anksčiau aprašytu termistoriumi.

Šio termometro sujungimas yra paprastas: **GND** kištukas siejamas su Arduino plokštės GND kištuku, **Vdd** kištukas – su 5V kištuku, o **Data** – su pasirinktu skaitmeniniu kištuku. Taip pat reikės panaudoti 4.7k Ω varžą tarp Vdd ir Data kištukų.



11 pav. Arduino ir DS18B20 sujungimo schema

Kaip anksčiau minėta, galima jungti daug tokių daviklių į bendrą magistralę, svarbiausia, kad jie būtų sujungti nuosekliai (žr. 3 priedą).

```
#include <OneWire.h>
#include <DallasTemperature.h>

int pin = 2;

OneWire oneWire(pin);
DallasTemperature sensors(&oneWire);

void setup()
{
    Serial.begin(9600);

    sensors.begin();
}

void loop()
{
    sensors.requestTemperatures();

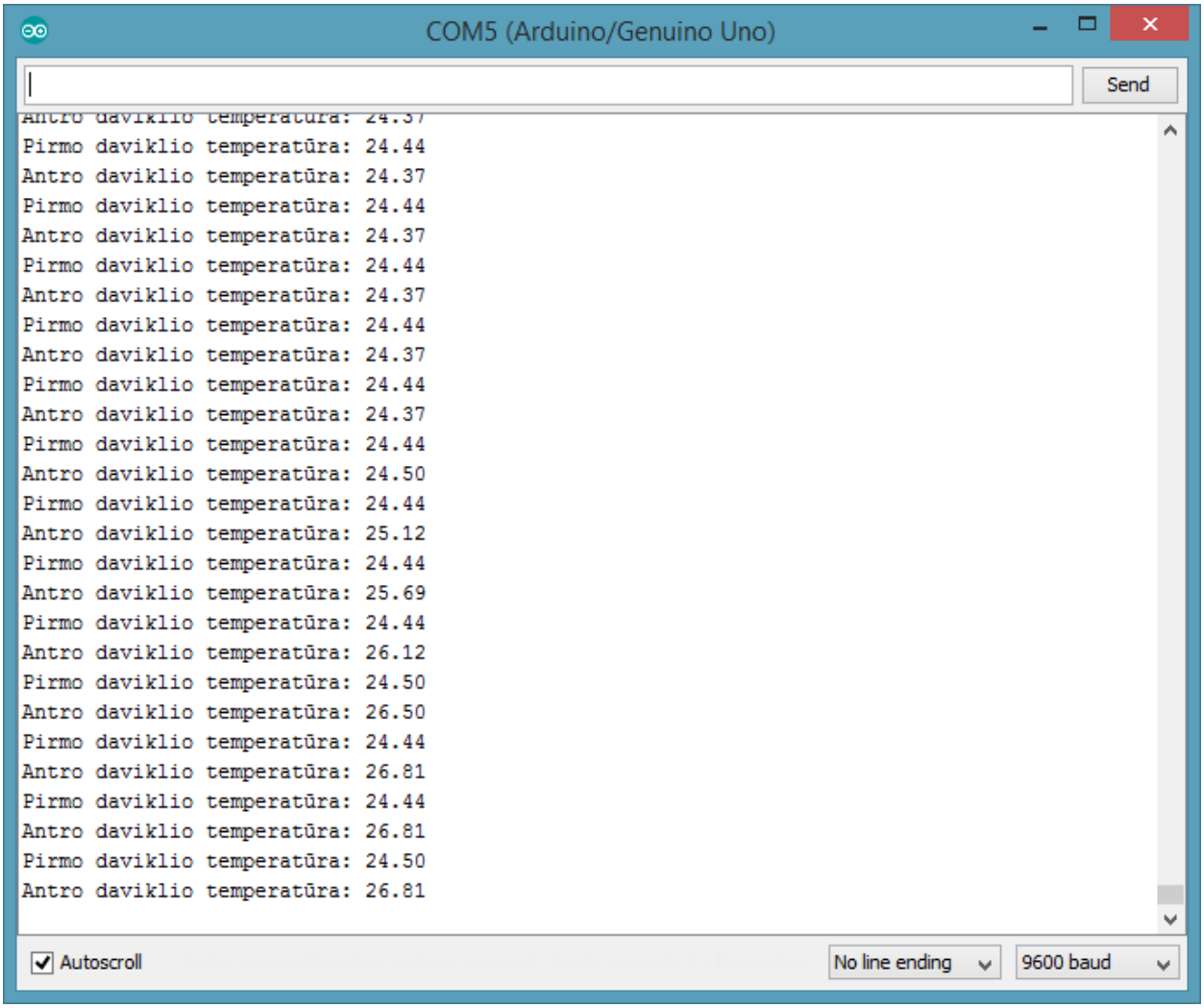
    Serial.print("Pirmo daviklio temperatūra: ");
    Serial.println(sensors.getTempCByIndex(0));

    Serial.print("Antro daviklio temperatūra: ");
    Serial.println(sensors.getTempCByIndex(1));

    delay(1000);
}
```

12 pav. DS18B20 ezkizas

Eskize įtraukiame kelias bibliotekas: OneWire ir DallasTemperature. Pirmoji skirta užmegzti ryšiui su davikliu. Pastaroji suteikia funkcionalumą gauti temperatūrą nedarant skaičiavimų patiems. Apsirašome OneWire bibliotekos instanciją ir perduodame jos nuorodą DallasTemperature bibliotekai. *loop()* funkcijoje pareikalaujame temperatūrų iš visų sujungtų daviklių ir tada atspausdiname pirmo ir antro daviklio temperatūras (13 pav.).



13 pav. DS18B20 eskizo rezultatai

5.8 Termistoriaus ir skaitmeninio termometro palyginimas

Prieš naudojant termistorių reikėjo sužinoti, kas tai yra, kaip veikia analoginis-skaitmeninis keitiklis, sužinoti, kas yra Steinhart-Hart'o koeficientai ir kaip juos apskaičiuoti naudojantis duomenų lapu ir termistoriaus kalkuliatoriumi. Reikėjo parašyti funkciją, paverčiančią gautą reikšmę į varžą, ir funkciją, paverčiančią varžą į temperatūrą.

Temperatūros gavimui skaitmeniniu termometru reikėjo sužinoti, kaip jį prijungti, o programinėje dalyje – įtraukti dvi bibliotekas, skirtas ryšio užmezgimui su termometru bei temperatūros gavimui, ir šių bibliotekų metodų pagalba gauti temperatūrą.

Atsižvelgiant į naudojimo paprastumą ir į tai, jog viename avilio modulyje gali būti daugiau nei vienas daviklis, sistemos kūrimui buvo pasirinktas skaitmeninis termometras. Taip pat šis daviklis

pranašesnis tuo, kad galima daug šių daviklių sujungti į bendrą magistralę, ir, naudojantis vienu kištuku, gauti informaciją iš visų daviklių. Skaitmeninių termometrų nereikia jungti į atskiras elektros grandines, todėl išnaudojama mažiau plokštės kištukų.

6. Avilio monitoringo sistema

Šiame skyriuje bus aprašyta autoriaus sukurta avilio monitoringo sistema, jos privalumai ir trūkumai, bei apžvelgtas sistemos funkcionalumas.

Sistema susideda iš trijų dalių: Arduino eskizo, Web aplikacijos ir konsolinės aplikacijos.

6.1 Arduino eskizas

```
#include <ArduinoJson.h>
#include <OneWire.h>
#include <DallasTemperature.h>

#define digitalPin 2

OneWire oneWire(digitalPin);
DallasTemperature sensors(&oneWire);

DeviceAddress davikliai[4] = {
    { 0x28, 0xFF, 0x24, 0x68, 0xA8, 0x15, 0x01, 0xBD },
    { 0x28, 0xFF, 0xEC, 0x66, 0xA8, 0x15, 0x03, 0x08 },
    { },
    { }
};

StaticJsonBuffer<100> jsonBuffer;
JsonObject& root = jsonBuffer.createObject();

String t;

void setup() {
  Serial.begin(9600);

  sensors.begin();
}

void loop() {
    sensors.requestTemperatures();

    t = "";

    for (int i = 0; i < 4; i++) {
        t = "t" + String(i);

        if (sensors.requestTemperaturesByAddress(davikliai[i])) {
            root[t] = sensors.getTempC(davikliai[i]);
        } else {
            root[t] = 0;
        }
    }

    root.printTo(Serial);
    Serial.println();
    Serial.flush();
}
```

14 pav. Sistemos dalis: eskizas

Eskize yra naudojamos trys bibliotekos: OneWire ir DallasTemperature, skirtos darbui su davikliais, ir ArduinoJson, suteikianti galimybę suformuoti JSON objektą (15 pav.) ir jį perduoti nuosekliuoju prievadu.

Toliau yra aprašomas Arduino kištukas, kurį naudos davikliai, sukuriama daviklio bibliotekos instancija bei aprašomi naudojamų daviklių adresai (kiekvienas daviklis turi savo unikalų adresą) bei sukuriama simbolių eilutė, kuri bus naudojama JSON objekto formavime. Toliau yra atidaromas nuoseklusis prievadas ir iškviečiamas metodas, paleidžiantis daviklio biblioteką. *loop()* funkcijoje yra iškviečiamas metodas, kuris paprašo informacijos iš daviklių. Toliau cikle kiekvienam davikliui yra nusiunčiama užklausa, ar tas daviklis egzistuoja. Jeigu taip, tai jo temperatūra nuskaitoma ir pridedama prie JSON objekto. Jeigu ne – įrašomas nulis. Galiausiai JSON objektas yra nusiunčiamas į nuoseklųjį prievadą.

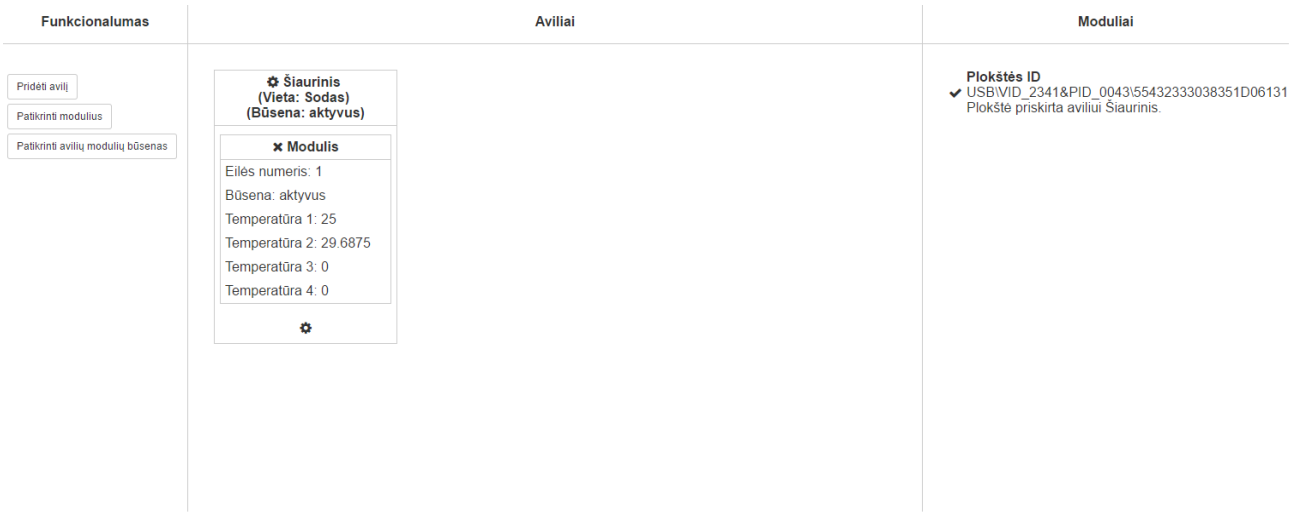
```
"{
  "t0": 24.625,
  "t1": 24.625,
  "t2": 0,
  "t3": 0
}"
```

15 pav. Siunčiamo JSON objekto pavyzdys

6.2 Web aplikacija

Web aplikacija buvo rašyta su Visual Studio programine įranga. Ją rašant buvo panaudotos šios technologijos:

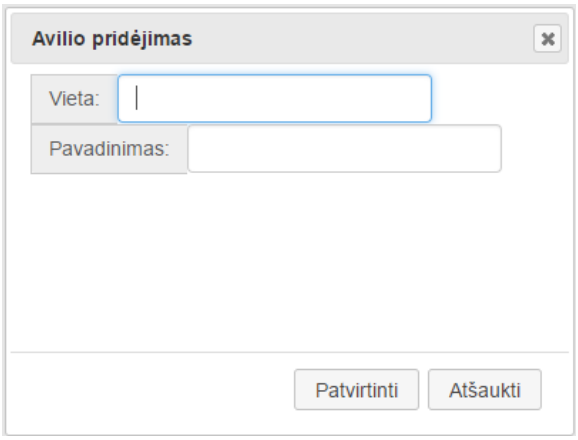
- Vidinei pusei (angl. *Back-end)* - .NET MVC karkasas ir C# programavimo kalba
- Išorinei pusei (angl. *Front-end)* arba vartotojo sąsajai – HTML ir Razor ženklinimo (angl. *Markup)* kalbos, CSS kalba ir BootStrap biblioteka – puslapio stiliaus taisyklėms aprašyti, JavaScript kalba ir JQuery biblioteka – puslapio funkcionalumui įgyvendinti, AlertifyJs biblioteka – atoveiksmių pranešimams rodyti, FontAwesome biblioteka - ikonoms.



16 pav. Web aplikacijos vartotojo sąsaja

Aplikacijos funkcionalumas:

- Pridėti avilį – ši funkcija suteikia vartotojui galimybę pridėti avilį, o tai darant – skirti vietą, kurioje jis stovi ir avilio pavadinimą.



17 pav. Avilio pridėjimo langas

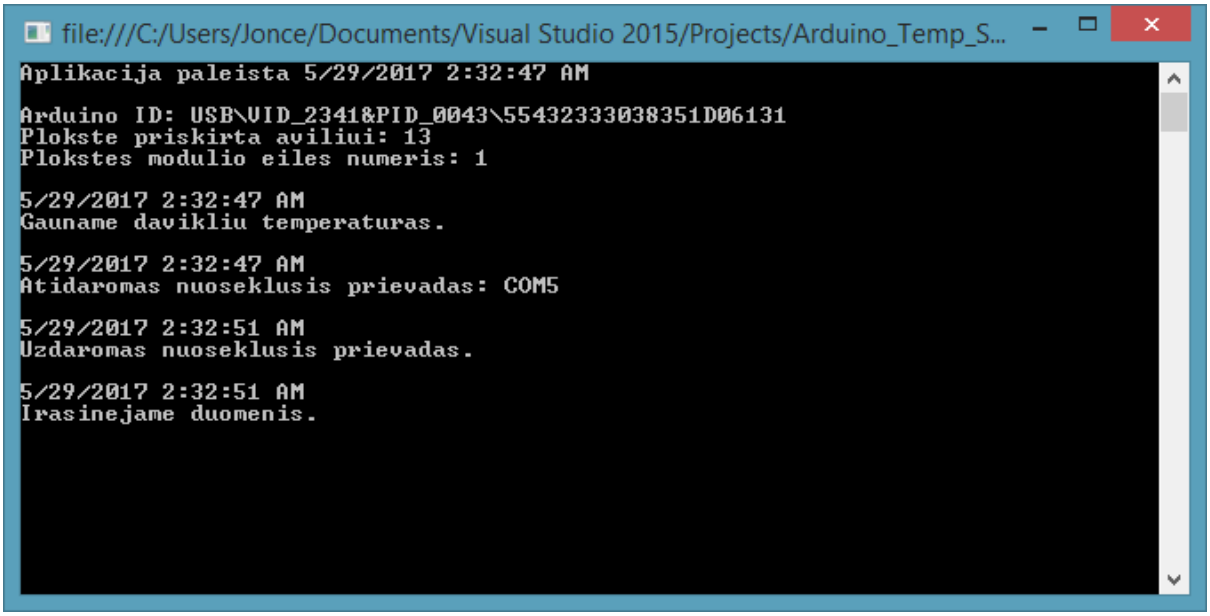
- Patikrinti modulius – funkcija, patikrinanti, ar yra prijungtų Arduino plokščių. Jeigu yra – atsiranda modulių stulpelyje su galimybe plokštę priskirti moduliui. Jeigu plokštė jau priskirta – rodoma, kuriame avilyje yra modulis su ta plokšte, todėl plokštės neįmanoma priskirti keliems aviliams.
- Patikrinti modulių būsenas – funkcija, patikrinanti, ar viskas gerai su aviliams priskirtomis plokštėmis. Patogi naudoti, kai kyla abejonių dėl temperatūros rodmenų.

Avilys gali būti vienos iš dvejų būsenų – aktyvus arba neaktyvus. Avilį galima padaryti neaktyviu paspaudus ⚙ ikoną. Padarius avilį neaktyviu, jo moduliai taip pat tampa neaktyviais. Neaktyvūs moduliai nerodo temperatūros rodmenų.

Moduliai gali būti panaikinti po vieną, paspaudus ✖, ir panaikinti visi, paspaudus ⚙.

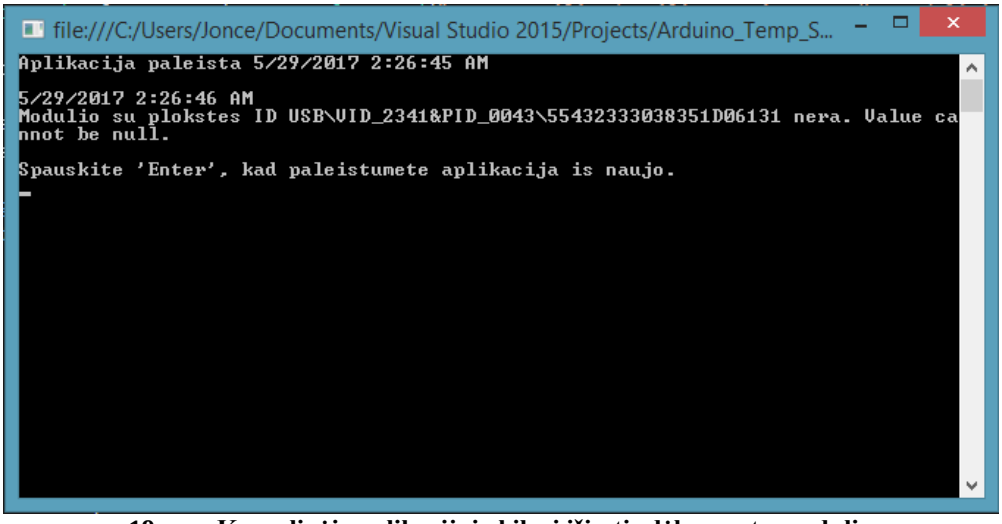
6.3 Konsolinė aplikacija

Konsolinė aplikacija yra atsakinga už temperatūros rodiklių duomenų įrašymą į duomenų bazę. Aplikacija patikrina, ar yra modulis su pateiktu Arduino plokštės ID, jeigu yra – ar tas modulis aktyvus. Toliau yra atidaromas nuoseklusis prievadas ir nuskaitomi duomenys, kuriuos siunčia Arduino. Gale vyksta duomenų įrašymas.



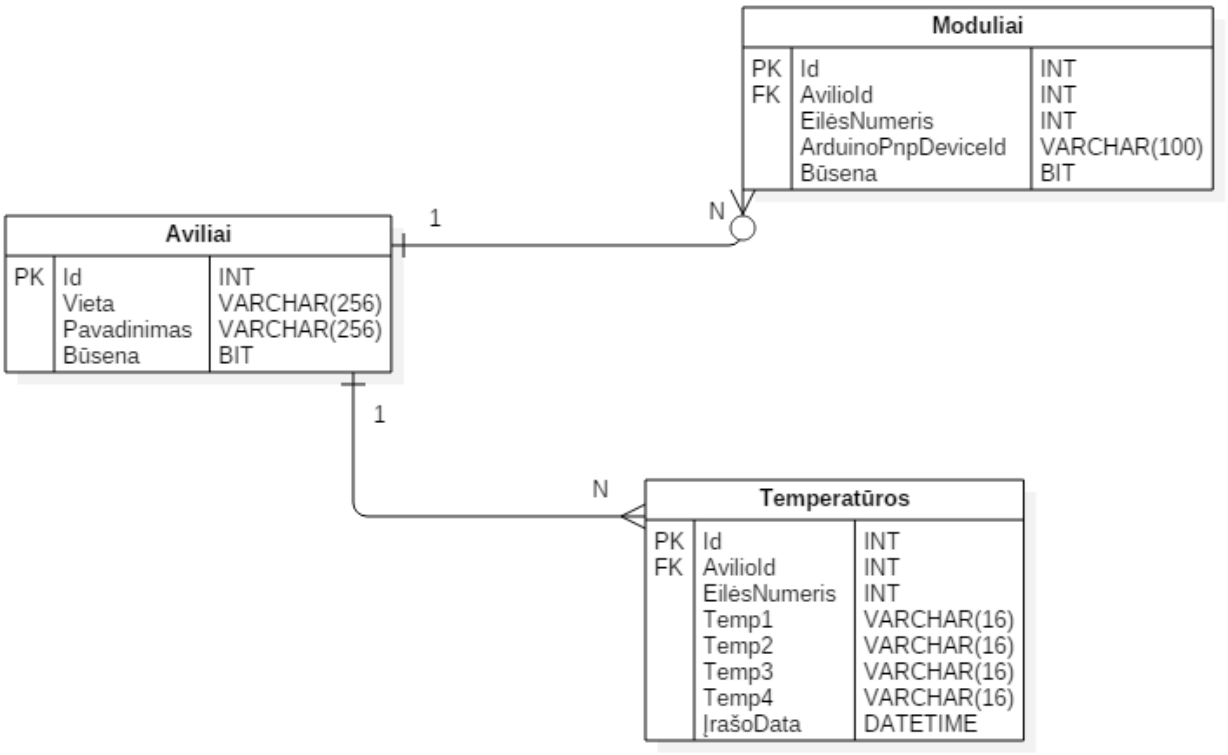
18 pav. Konsolinė aplikacija

Aplikacija nėra pilnai automatinė – išimties (angl. *Exception)* atveju reikia žmogaus prisilietimo, kad išspręstų kilusią bėdą. Stengtąsi apdoroti dažniausias išimtis, pvz., neegzistuojantis nuoseklusis prievadas arba nerastas modulis su pateiktu Arduino ID (19 pav.).



19 pav. Konsolinėje aplikacijoje kilusi išimtis dėl nerasto modulio

6.4 Esybių ryšių diagrama



20 pav. Esybių ryšių diagrama

Rezultatai ir išvados

Šiame darbe buvo apžvelgtas Langstroth‘o avilys, jo struktūra ir ypatumai. Taip pat buvo apžvelgti rankinis ir elektroninis monitoringai, jų trūkumai ir privalumai, bei kokius parametrus galima stebėti. Dokumente taip pat buvo peržiūrėta vokiečių programuotojo Maximilian‘o Michels‘o sukurta avilio monitoringo sistema. Gale buvo sukurta ir įvertinta monitoringo sistema.

Viso darbo metu buvo gauti tokie rezultatai:

1. Apžvelgtas Langstroth‘o avilys ir jo struktūra. Šio avilio privalumai yra išimami rėmai ir tarprėmio – tam tikro atstumo, kuriuo bitės lipdo korius nuo kito – panaudojimas.
2. Apžvelgti stebimi parametrai ir kokias žinias suteikia jų informacija, dėl kokių priežasčių jie svarbūs. Vienas svarbiausių ir lengviausiai pamatuojamų parametrų yra temperatūra. Temperatūros pokyčiai avilyje gali būti indikacija apie spiečiaus būklę.
3. Peržiūrėta viena avilio monitoringo sistema ir apžvelgtos joje naudojamos technologijos. Svorio stebėjimui autorius naudojo „PCE-PCS 30“ svarstykles, temperatūros – „DS1820“ skaitmeninį termometrą, aplinkos oro monitoringui - „WS-2310“ meteorologinę stotį. Padarytos išvados, kad naudotos svarstyklės netinka monitoringui dėl per mažos svėrimo galios, veikimo per mažoje temperatūrų riboje ir nepatogaus laidų kiekio. Temperatūros daviklis gali būti naudojamas monitoringui dėl plačios temperatūrų ribos ir galimybės naudoti kelis tokius daviklius vienu metu.
4. Sudaryta modulinio avilio monitoringo sistemos principinė schema, nustatyti tokie privalumai kaip mobilumas ir paprastas naudojimas.
5. Parašytas programinis kodas termistoriui IM120628010 ir skaitmeniniui termometrui DS18B20, praktiškai pamatuota temperatūra. Davikliai buvo palyginti ir įvertinti – tiek iš programinės dalies, tiek iš techninės įrangos pusių.
6. Sukurtas monitoringo sistemos prototipas, susidedantis iš trijų dalių: Arduino eskizo, surenkančio duomenis iš daviklių ir perduodančių jį į nuoseklųjį prievadą, Web aplikacijos ir konsolinės aplikacijos, atliekančios duomenų įrašymo į duomenų bazę funkciją. Didžiausias sistemos minusas yra nuosekliojo prievado naudojimas – vienu metu juo galima atlikti tik vieną veiksmą, todėl šiuo atveju reikia rinktis, ar rodyti temperatūrą web aplikacijoje, ar

įrašinėti duomenis. Taip pat, atskirai plokštei reikėtų atskiros konsolinės aplikacijos, nes gali skirtis nuosekliųjų prievadų konfigūracija.

Atsižvelgiant į rezultatus yra daromos tokios išvados:

- Skaitmeninis termometras yra pranašesnis už termistorių naudojimo paprastumu. Egzistuojančios kodo bibliotekos nuima skaičiavimų rašymo naštą, o galimybė vienu kištuku gauti duomenis iš kelių daviklių sumažina reikalingos įrangos apimtį.
- Nuosekliojo prievado naudojimas yra ribotas – šiuo atveju kiekviena plokštė jungiasi USB kabeliu, o tai reiškia, kad kompiuteris turi būti prie avilio. Įgyvendinus kitokį duomenų perdavimo būdą reikėtų perdaryti dalį aplikacijos.
- Nėra bendro funkcionalumo daugiau negu vienai plokštei – kiekvienai plokštei reikėtų atskirai konfigūruoti nuosekliojo prievado nustatymus, dėl to reikėtų ir atskirų konsolinių aplikacijų norint įrašinėti duomenis.
- Temperatūros matavimas gali būti įgyvendintas su Arduino plokšte ir skaitmeniniais termometrais. Parašyta aplikacija suteikia pradinį funkcionalumą darbui su aviliais ir jų moduliais, tačiau turėtų būti pasirinktas efektyvesnis duomenų perdavimo būdas ir, atsižvelgiant į pasirinktą būdą, perrašytos darbo su duomenimis dalys.

Literatūros sąrašas

[Bla09] Howland Blackiston. *Beekeeping For Dummies. 2nd Edition*. Indianapolis: Wiley Publishing, 2009.

[Bur14] Gates N. Burton. The temperature of the bee colony, United States Department of Agriculture, Dept. Bull. No. 96. 1914

[Cig06] Dale Cigoy. „Basics of Temperature Measurement: Thermistors". In: *Keithley Instruments, Inc. No. 2765,* 2006.

[Cra79] Eva Crane. „Bees and beekeeping in the tropics, and trade in honey and beeswax with special reference to the commonwealt". In: *Beekeeping in rural development*. The Commonwealth Secretariat, 1979, p. 1-19.

[Dou76] Keith M. Doull. „The effects of different humidities on the hatching of the eggs of honeybees". In: *Apidologie 7 (1)*. Berlin: Springer Verlag, 1976, p. 61-66.

[Eva15] Sandra K. Evans. „Electronic beehive monitoring – applications to research". In: *Julius-Kühn-Archiv 450*. Berlin: Arno Brynda, 2015.

[FRR+96] Jm. Flores, Ja. Ruiz, Jm. Ruz, F. Puerta, M. Bustos, et al.. „Effect of temperature and humidity of sealed brood on chalkbrood development under controlled conditions". In: *Apidologie 27(4).* Berlin: Springer Verlag, 1996, p. 185-192.

[FSp58] J. B. Free, Yvette Spencer-Booth. „Observations on the Temperature Regulation and Food Consumption of Honeybees (*Apis Mellifera*)". In: *Journal of Experimental Biology, Vol. 35*. London: Cambridge University Press, 1958, p. 930-937.

[Kri10] Gene Kritsky. *The quest for the perfect hive: a history of innovation in bee culture*. , New York: Oxford University Press, 2010.

[KV97] B. Kraus, H.H.W. Velthuis. „High Humidity in the Honey Bee (*Apis mellifera L.*) Brood Nest Limits Reproduction of the Parasitic Mite (*Varroa jacobsoni*) Oud“. In: *Naturwissenschaften 84*. Berlin: Springer Verlag, 1997, p. 217–218.

[MH15] W.G. Meikle, N. Holst. „Application of continuous monitoring of honeybee colonies“. In: *Apidologie 46 (1)*. Paris: Springer Verlag, 2015, p. 10-22.

[Mic11] Maximilian Michels. *A Beehive Monitoring System Incorporating Optical Flow as a Source of Information*. 2011.

[See74] Thomas D. Seeley. „Atmospheric carbon dioxide regulation in honey-bee (*apis mellifera*) colonies“. In: *Journal of Insect Physiol*, 1974, Vol. 20, p. 2301-2305.

[SM97] Henry E. Sostmann, Philip D. Mets. „Fundamentals Of Thermometry. Part VI: Thermistor Thermometers”. In: *Isotech Journal Of Thermometry, Vol. 8 No. 2,* 1997, p.66-123.

[Sra02] Justinas Sraigis. *Mėgėjiškoji bitininkystė*. Vilnius: Margi raštai, 2002.

[ZSM12] A. Zacepins, E. Stalidzans, J. Meitalovs. „Application of information technologies in precision apiculture“. In: *Proceedings of the 13th International Conference on Precision Agriculture*, Indianapolis, USA, 2012.

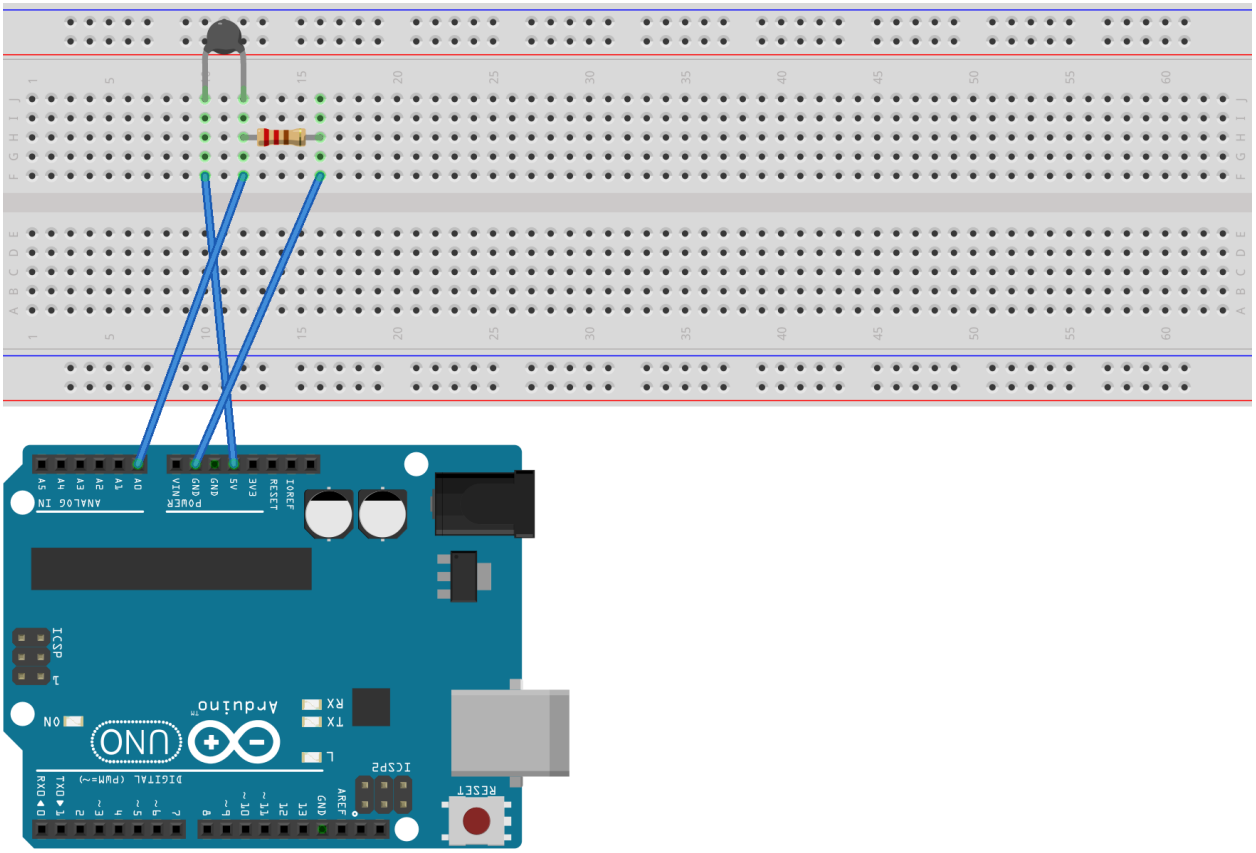
Priedai

1 priedas. IM120628010 termistoriaus varžų ir atitinkamų temperatūrų lentelė.

R (25℃) : 10KΩ B(25℃/50℃):3950

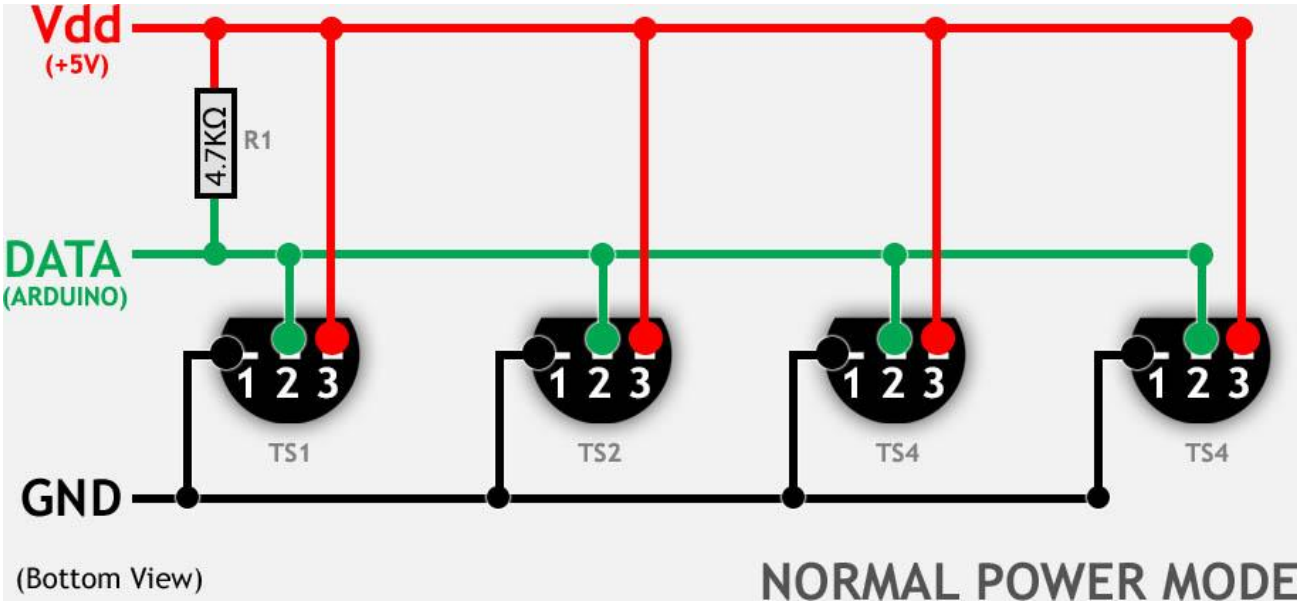
T (℃)	R (kΩ)	T (℃)	R (KΩ)	T (℃)	R (kΩ)	T (℃)	R (kΩ)
-25	133.500	15	15.7511	55	2.96331	95	.781670
-24	125.672	16	15.0306	56	2.85569	96	.758701
-23	118.350	17	14.3472	57	2.75256	97	.736519
-22	111.498	18	13.6987	58	2.65369	98	.715094
-21	105.084	19	13.0833	59	2.55890	99	.694397
-20	99.0773	20	12.4990	60	2.46799	100	.674400
-19	93.4469	21	11.9441	61	2.38080	101	.655075
-18	88.1750	22	11.4169	62	2.29714	102	.636398
-17	83.2296	23	10.9161	63	2.21686	103	.618345
-16	78.5909	24	10.4400	64	2.13980	104	.600890
-15	74.2384	25	10.0000	65	2.06583	105	.584013
-14	70.1527	26	9.55693	66	1.99480	106	.567690
-13	66.3162	27	9.14743	67	1.92658	107	.551902
-12	62.7122	28	8.75777	68	1.86105	108	.536629
-11	59.3254	29	8.38690	69	1.79809	109	.521852
-10	56.1416	30	8.03380	70	1.73758	110	.507552
-9	53.1475	31	7.69753	71	1.67942	111	.493712
-8	50.3307	32	7.37721	72	1.62351	112	.480316
-7	47.6799	33	7.07200	73	1.56975	113	.467346
-6	45.1842	34	6.78110	74	1.51804	114	.454788
-5	42.8339	35	6.50378	75	1.47300	115	.442627
-4	40.6197	36	6.23934	76	1.42045	116	.430848
-3	38.5330	37	5.98711	77	1.37439	117	.419438
-2	36.5656	38	5.74646	78	1.33007	118	.408384
-1	34.7103	39	5.51680	79	1.28740	119	.397674
0	32.9600	40	5.29758	80	1.24632	120	.387294
1	31.3081	41	5.08828	81	1.20675	121	.377233
2	29.7487	42	4.88838	82	1.16864	122	.367481
3	28.2760	43	4.69743	83	1.13193	123	.358026
4	26.8848	44	4.51498	84	1.09655	124	.348859
5	25.5702	45	4.34060	85	1.06246	125	.339968
6	24.3274	46	4.17391	86	1.02960		
7	23.1523	47	4.01452	87	.997924		
8	22.0407	48	3.86207	88	.967376		
9	20.9889	49	3.71624	89	.937916		
10	19.9934	50	3.58800	90	.909498		
11	19.0509	51	3.44314	91	.882083		
12	18.1582	52	3.31529	92	.855630		
13	17.3124	53	3.19287	93	.830101		
14	16.5109	54	3.07563	94	.805459		

2 priedas. Arduino ir bandomosios plokštės sujungimo schema[3].



[3] Schema buvo braižyta su atviro naudojimo programa **Fritzing**.

3 priedas. Kelių DS18B20 termometrų jungimo schema[4]. Prieiga internetu: https://www.tweaking4all.com/hardware/arduino/arduino-ds18b20-temperature-sensor/



[4] Schema buvo braižyta su atviro naudojimo programa **Fritzing**.