

VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

INFORMATIKOS INSTITUTAS

KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Bakalauro baigiamasis darbas

Bičių avilio monitoringas

Atliko:

Ernestas Kaukėnas parašas

Darbo vadovas:

Dr. Asist. Tomas Plankis

Vilnius

2019

Santrauka **4**
Summary **5**
Įvadas **6**
1. Bičių avilio monitoringas **7**
1.1. Tradicinis avilio stebėjimo ir tikrinimo metodas 7
1.2. Bičių avilio stebėjimas jutikliais 8
1.2.1. Avilio temperatūros stebėjimas 8
1.2.2. Avilio svorio stebėjimas 9
1.2.3. Medaus grynumo stebėjimas 10
1.2.4. Bičių judėjimo stebėjimas 10
1.2.5. Išorinių veiksnių stebėjimas 10
1.3. Jutiklių diegimas į avilius 11
2. Arduino ir maketavimo plokštės apžvalga **12**
2.1. Arduino plokštė 12
2.2. Programinė įranga „ARDUINO“ 13
2.3. Maketavimo plokštė 14
3. Temperatūros jutikliai **15**
3.1. Temperatūros stebėjimas infraraudonaisiais jutiklis 15
3.2. Temperatūros matavimas analoginiu termistoriu 16
3.3. Temperatūros matavimas skaitmeniniu DS18B20 jutikliu 17
3.4. Temperatūros jutiklių tarpusavio apibendrintas palyginimas 18
4. Svorio Jutiklis **20**
4.1. Hx711 apkrovos jutiklis 20
4.2. Hx711 stiprintuvas 21
4.3. Svarstyklių kalibravimas ir svorio matavimas 22
5. Bičių avilio monitoringo sistemos kūrimas **24**
5.1. Avilio duomenų rinkimas ir siuntimas 24
5.2. Surinktų duomenų saugojimas 24
5.3. Web aplikacija duomenims atvaizduoti 25
Išvados ir rekomendacijos **27**
Ateities tyrimų planas **28**
Literatūros sąrašas **29**
PRIEDAS A **30**
PRIEDAS B **31**

PRIEDAS C 32
PRIEDAS D 33

Santrauka

Darbo tikslas yra išanalizuoti, kokius bičių avilio parametrus galima stebėti su elektroniniais jutikliais, bei sukurti bičių avilio stebėjimo sistemą, kuri rinktų duomenis, juos saugotų ir atitinkamai surinktus duomenis atvaizduotų diagramose. Tikslui įgyvendinti yra analizuojama, kaip yra stebimas avilys tiesioginių būdu, bei pasitelkus elektroninius jutiklius, nurodomi stebėjimo privalumai ir trūkumai. Aprašoma, kokius avilio parametrus yra geriausia stebėti, kad išgauti kuo daugiau naudingos informacijos apie avilį, bei kokius jutiklius reikia pasitelki tam tikrų parametrų stebėjimui. Didžiausias dėmėsys yra skiriamas bičių avilio temperatūros ir svorio stebėjimui, aprašomi jų veikimo pincipai, bei yra palyginami skirtingi temperatūros stebėjimo jutiklių tipai. Praktinėje dalyje yra sukurta bičių avilio stebėjimo sistema, skirta stebėti ir surinkti duomenis apie avilio temperatūrą ir svorį, bei tuos duomenis atvaizduoti diagramose.

Raktiniai žodžiai: Arduino plokštė, bičių avilio monitoringas, termistorius, temperatūros jutiklis, svorio jutiklis, temperatūros stebėjimas, svorio stebėjimas

Summary

Beehive Monitoring

In this paper author describes the main devices and parameters for bee hive monitoring. To meet main goal, author reviews traditional beehive monitoring, compares it to the monitoring devices, concludes their advantages and disadvantages. Work also presents, which beehive parameters should gather most attention and what important data we can collect from it, mainly focusing on temperature and weight devices, compares different types of devices for temperature measurement. At the end, author creates prototype for monitoring bee hive temperature and weight, collecting its data and storing it to database, which is also is being sent to server to be displayed as a graphs.

Keywords: Arduino board, beehive monitoring, thermistor, temperature sensors, load cells, temperature monitoring, weight monitoring

Įvadas

Šiuo metu bičių tema yra labai aktuali ir kritinė globaliniu mastu – jos yra vienas svarbiausių organizmų, palaikančių augalų ekosistemą ir jų gyvenimo ciklą dėl gebėjimo apdulkinti žydinčius augalus. Deja, dėl pilnai nežinomų priežasčių, bitės išgyvena krizę - kolonijos nykimo sindromą (angl. „*CCD or Colony Collapse Disorder*“), dėl ko bičių kolonijos miršta visame pasaulyje. Dėl didelių nuostolių ekosistemai ir agrikultūrai, pradėta imtis įvairiausių priemonių ir metodų, kaip galima būtų stebėti ir nustatyti šio fenomeno priežastis, bei tas priežastis pašalinti, tuo pačiu nepakenkus pačioms bitėms. Tam į pagalbą galima pasitelkti elektroninius įrenginius. Pasitelkus elektroninius įrenginius, skirtus stebėti bitėms aviliuose, galima išgauti kuo daugiau naudingos informacijos apie bites ir kuo mažiau arba visiškai nepakenkiant bitėms, nes be technologijų, bitininkai privalo atidarinėti avilius tiesiogiai, kas sukelia stresą pačioms bitėms. Dėka XXI a. amžiaus technologijų, bitininkai turi labai platų, kokybišką ir pigų būdą stebėti ir analizuoti bites su tam skirtais jutikliais, bei turi pakankamai platų pasirinkimą kaip tai įgyvendinti ir patenkinti savo lūkesčius: naudojant skaitmeninius arba analoginius jutiklius, belaidę arba laidų sujungtą tinklo įrangą, duomenų bazes arba šviesos diodų (angl. *„LED or Light-emitting diode* “) ekranus duomenims atvaizduoti [1].

Darbo tikslas yra išanalizuoti jutiklius, skirtus stebėti avilio būseną, palyginti jutiklius tarpusavyje, pažiūrėti kokius avilio parametrus jie stebi. Praktinėje dalyje iškelta užduotis suprogramuoti temperatūros ir svorio jutiklius, kad būtų galima stebėti bičių avilio temperatūrą ir svorį, juos saugoti bei atitinkamai atvaizduoti diagramomis. Darbo tikslui įgyvendinti, buvo atlikti šie uždaviniai:

1. Apžvelgtas tiesioginis avilio tikrinimo būdas, bei nurodyti jo trūkumai – iš kurių svarbiausias yra avilio tikrinimas žiemą, kas dažnai gali pražudyti visą koloniją.

2. Apžvelgti svarbiausi stebimi parametrai ir atskirai yra pakomentuoti, kokiais jutikliais juos galima stebėti. Aprašoma, kokią informaciją nusako matuojami parametrai apie bičių avilį.

3. Aprašyti ir palyginti temperatūros stebėjimo jutikliai. Tarpusavyje palyginti termistoriai, skaitmeniniai ir infraraudonieji davikliai, nurodyti jų naudojimo privalūmai ir trūkumai.

4. Aprašytas HX711 svarstyklių jutiklis, jo veikimo principas, aprašytas svarstyklių modulis, kuris skirtas konvertuoti svarstyklių matuojamą varžą į skaitmeninę įtampą.

5. Sukurta sistema, kuri renka temperatūros ir svorio duomenis, juos siunčia į serverį ir tuos duomenis atvaizduoja diagramomis.

1. Bičių avilio monitoringas

Šiame skyriuje yra aprašyti tradicinis ir elektroninis avilio stebėjimo būdai, kokius parametrus galima stebėti su tam skirtais elektroniniais jutikliais.

1.1. Tradicinis avilio stebėjimo ir tikrinimo metodas

Geras ir patyręs bitininkas pasitelkia kelis faktorius, kad nustatytų kada bičių avilį reikia tikrinti. Dažniausiai tai jis nustato, pasitelkdamas uoslę ir garsą - sveikas ir tvarkingas avilys visada kvepia gerai, jis neturi skleisti keisto arba smirdančio kvapo, sveikame avilyje bičių skleidžiamas garsas turi būti ramus, kas parodo, kad bitės yra ramioje būsenoje. Tačiau vien šių faktorių yra negana, kad būti 100 procentų įsitikinus apie avilio būseną, todėl jį bet kokiais atvejais anksčiau ar vėliau reikia atidaryti ir patikrinti jo būseną iš vidaus. Aviliai dažniausiai pradedami tikrinti pavasarį, kai prasideda augalų žydėjimo sezonas (pirmas patikrinimas visada prasideda nuo avilio paruošimo naujai arba peržiemojusiai bičių kolonijai). Ne sezono metu, tai yra žiemą, bitininkas tikrina, ar bičių šeima sugebės sėkmingai peržiemoti (šiam tikrinimui bitininkas gali pasitelkti tik garsą, nes atidarius avilį būtų pražudyta visa bičių šeima).

Tradicinio avilio vidaus tikrinimo metu, bitininkas fiziškai atidaro bičių avilį ir pilnai išrenka jo dalis, kad atliktų tam tikrus jam svarbius patikrinimus, tokius kaip avilio rėmų būsenos patikrą, ar bitės pakankamai juos užpildė medumi, patikrina ar avilys neperpildytas ir ar nereikia jo plėsti į viršų arba į šonus (plėtimas priklauso nuo avilio tipo: gulstiniai aviliai plečiami į šonus, daugiaaukščiai aviliai – į viršų). Taip pat bitininkas tikrina, ar aviliuose nėra užsiveisusių parazitų ar kenkėjų, kurie tiesiogiai kenkia bitėms arba aviliams. Toks avilio tikrinimo būdas sukelia labai daug streso bitėms, nes atidarant avilius bitininkas naudoja dūminę, bei pažeidžia jų lizdą, nes turi išardyti bičių sukurtą korių struktūrą, kieno pasekmėje bitės gali nusilpti ir tapti pažeidžiamos. Taip pat atidarytas avilys, nors ir neilgam, suteikia proga kenkėjams ir parazitams (tokiems kaip *Mažasis avilio vabalas*, kuris medžiojo avilius pagal kvapą ir deda savo kiaušinėlius į korį [2]) tiesiogiai pakenkti tiek bitėms, tiek pačiam lizdui.

Žiema bitininkas negali fiziškai atidarinėti avilio, kadangi tai reikštų visos avilio kolonijos žūtį, todėl prieš kiekvieną bičių žiemojimo sezoną, bitininkas privalo įsitikinti ir pasirūpinti, kad bitės žiemos sezoną peržiemotų sėkmingai. Tam bitininkas visada atlieka tolesnius darbus:

- Susiaurina lizdą, kadangi žiemą nenaudojami rėmai arba koriai šaldo avilį iš vidaus.
- Palieka pakankamai maisto bitėms. Cukraus sirupu bitės pradedamos maitinti rudenį (sirupas supilamas į bičių penėtuves), o žiemą maitinamos (jeigu trūksta maisto) kandžia tešla (pastą primenanti cukraus masė).
- Užlopyti didesnes avilio angas, kad apsaugoti avilį nuo skersvėjo (mažesnes angas bitės pačios užakiuoja pikiu).

1.2. Bičių avilio stebėjimas jutikliais

Bičių avilio monitoringo pagrindinis tikslas yra automatizuotas duomenų rinkimas apie bičių šeimą tam skirtais jutikliais. Šis stebėjimo būdas bitininkams suteikia pastovią ir tikslią informaciją apie bičių šeimos būklę, neatidarant avilio ir nesukeliant diskomforto pačioms bitėms, kas yra ypač svarbu žiemos periodu. Bitininkas gali pats pasirinkti kokius parametrus jis nori avilyje stebėti ir su profesionalų pagalba arba pats savo žiniomis gali jutiklius įdiegti į bičių avilį, kurie vėliau per pasirinktus kanalus siųs informaciją bitininkui į elektroninius įrenginius arba tiesiog rodys ant įdiegtų ekranų (pvz. ant įdiegtų LCD ekranų).

1.2.1. Avilio temperatūros stebėjimas

Tam, kad būtų tinkama vystimuisi aplinka, bitės avilyje stengiasi palaikyti nuo 32°C iki 35°C laipsnių šilumos temperatūrą, nepriklausomai nuo esamo sezono (žiemos sezono metu, 32°C ~ 35°C temperatūra yra palaikoma tik pačiame šeimos centre, tuo tarpu išorinėse šeimos dalyse temperatūra gali būti mažesnė). Temperatūros kitimas, kad ir menkas (užtenka 0.5°C), turi didelį poveikį bičių sveikatai ir vystymosi raidai – viršijant 35°C medaus kokybė pradeda kristi ir atsiranda didelė tikimybė, kad naujagimės bitės nebeišgyvens[3]. nukritus temperatūrai žemiau optimalios ribos, atsiranda tikimybė, jog visos bitės išmirs. Bitininkas turi jau pradėti reaguoti, kai yra pastebimas temperatūros šokinėjimas avilyje (žr. 1 pav.).



1 pav. Bičių avilio temperatūros nestabilumo užfiksavimas

Su jutiklių pagalba, bitininkas gali lengvai stebėti temperatūros kitimą avilyje ir daryti tam tikras išvadas. Tai ypač palengvina bitininko darbą, nes analizuojant gautą informaciją, galima daryti tokias prielaidas [4]:

1. Avilys turi arba neturi bičių motinėlės.
2. Bitės yra ramioje arba įniršio būsenoje (angl. „*Broodless state*“), kas gali būti dėl sezono daromos įtakos (pvz. dėl žiemos).
3. Pas bites prasidėjo spietimosi (angl. „*Swarming*“) sezonas, kuriuo metu jos pradeda ieškoti naujos vietos arba naujų avilių bičių koriams lipdyti.

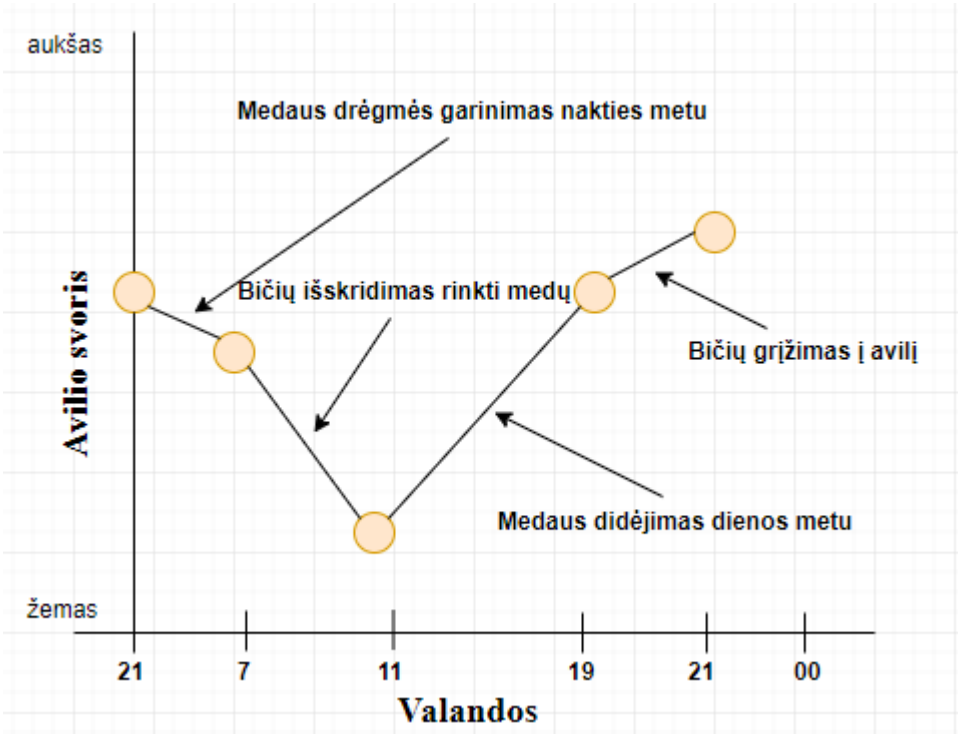
Temperatūros stebėjimui bitininkas gali rinktis tarp skirtingų temperatūros jutiklių tipų – analoginių, skaitmeninių, bei rečiau naudojamų, bet sparčiai populiarumą įgijančių – infraraudonųjų spindulių. Pagrindinis skirtumas tarp šių jutiklių yra tas, kad skaitmeninius

jutiklius yra žymiai lengviau įdiegti į avilį negu analoginius, tačiau skaitmeniniai jutikliai yra brangesni, bei turi didesnę paklaidą, tuo tarpu infraraudonųjų spindulių jutiliai yra diegiami avilio išorėje. Detaliau apie šių jutiklių privalumus ir veikimo principus bus aptarta vėlesniuose (3.1, 3.2, 3.3) šio darbo skyriuose.

1.2.2. Avilio svorio stebėjimas

Pastovus informacijos gavimas apie avilio svorio kitimą yra labai svarbus bitininkui, kadangi šie duomenys nurodo bičių avilio populiacijos didėjimą ir produktyvumą - kiek medaus bitės suneša sezono metu ir kiek jos suvartoja medaus žiemos metu. Medaus rinkimo (aktyvaus) sezono metu, avilio svorio kitimas turi sekančią eigą[5] (žr. 2 pav.):

- Anksti ryte būna matomas svorio mažėjimas, kadangi bitės išskrenda rinkti nektaro.
- Vėliau matomas svorio kilimas, kadangi nektarą renkančios bitės pradeda grįžti į avilį, bei medaus didėjimas iš praeitos dienos surinkto derliaus.
- Prasidėjus vakarui iki pat ryto matomas lėtas avilio svorio mažėjimas dėl bičių surinkto nektaro perdirbimo ir vartojimo, bei susikaupusios drėgmės šalinimo iš avilio.



2 pav. Avilio svorio kaita sezono metu[6]

Pasitelkus šią informaciją, bitininkas žino, kada jam pradėti rinkti medų, o žiemos atveju – ar jam reikia padėti bitėms peržiemoti, tai yra pamaitinti kandžia tešla(sirupas arba tirpintas cukrus yra nerekomenduojamas, kadangi savyje turi daug vandens, todėl bitėms prireiks daugiau pastangų, kad tą vandenį išgarinti). Dėl jutiklio svorio apribojimų, bitininkas turi įvertinti savo avilio dydį ir svorį, bei ateities planus, susijusius su šiuo aviliu (ar jis nesiruošia

avilio didinti) ir tik tada pasirinkti atitinkamus jutiklius (vidutiniškai pilni vidutinio dydžio aviliai sveria nuo 80 kilogramų) avilio svoriui matuoti[7].

1.2.3. Medaus grynumo stebėjimas

Pasitelkus optinio pluošto poslinkio jutiklius (angl. „*fiber optic displacement sensor*“), galima nustatyti, kokia yra gryno medaus koncentracija koriuose. Pagal išgaunamus duomenis, bitininkas gali matyti, kokią dalį medaus sudaro cukrus (fruktozė ir gliukozė). Taip pat šie jutikliai yra naudojami, kad stebėti pačio medaus brandumą koriuose (dažniausiai brandumas koriuose svyruoja nuo 80% iki 100%). Tai padeda nustatyti, kada galima pradėti kopti medų (kai medaus medaus brandumas koriuose pasiekia 100%). Be to, jutikliai dažnai yra naudojami analizuojant pačio medaus sudėtį: tikrinama ar medus nėra užkrėstas, tikrinamas jo natūralumas ir papildomi pridėti priedai (pvz. cukrus, vanduo)[8].

1.2.4. Bičių judėjimo stebėjimas

Pagal bičių judėjimą, jų atkeliavimo ir iškeliavimo iš avilio kryptį, galima nustatyti, ar avilys yra stiprus arba silpnas, bei iš kurių vietų ir kokių augalų bitės išgauna nektarą. Taip pat pagal judėjimo stebėjimą, galima pamatyti, kokios būsenos bitės atkeliauja į avilį – ar jos padengtos purvu, žiedadulkėmis ar sakais. Tai padeda bitininkui nustatyti trajektoriją arba tam tikrą vietovę, kur šio avilio bitės dažniausiai keliauja rinkti medų. Bičių judėjimo stebėjimui dažniausiai yra naudojami infraraudonosios LED lempos ir krūvio sąsajos įtaisų (angl.*„CCD – Charge-coupled device*“) kameros[9]. Bičių stebėjimas vyksta pagal sekantį metodą:

1. Bitininkas pasirenka atitinkamą bičių skaičių ir pažymį jas juodu simboliu arba kodavimo (angl. *„encoding“*) žyme.
2. CCD kamera pritvirtinama prie avilio apatinės lentos su įėjimu (angl. „*Bottom board with entrance cleat“*), dar kitaip vadinama laka ir yra įjungiama.
3. Kamera nufilmuoja vaizdą, bitei atskridus arba išskridus iš avilio.
4. Jeigu kamera atpažįsta juodą simbolį/kodavimo žymę – siunčia duomenis apie simbolį (bitę) į serverį arba kompiuterį.

Bitininkas pats pasirenka, ar jam žymėti bites tiesiog simboliu, ar susikurti visą masyvą skirtingų kodavimo žymių – priklausomai ar jam svarbu yra individualiai išskirti kiekvieną bitę ir sužinoti konkrečią informaciją apie ją (tada yra naudojamos kodavimo žymės), ar tiesiog sužinoti preliminarią bendrą informaciją apie visą koloniją (naudojamas pasirinktas simbolis).

1.2.5. Išorinių veiksnių stebėjimas

Be vidinės bičių avilio informacijos, bitininkui taip pat yra svarbu žinoti kas vyksta aplink avilio išorę. Dažniausiai didžiausias dėmesys yra kreipiamas į oro temperatūrą, bei drėgmę – šitie veiksniai daro didžiausią įtaka tiek bitėms, tiek pačiai šeimai, nes tada bitininkas mato, ar reikia pradėti galvoti apie bičių avilio apsaugą, ar dėl oro permainų reikia statyti kokias nors pavėsines. Išorės stebėjimui pasitelkiami labiau tradiciški ir buitiški jutikliai – meteorologinės

stotelės, kurios matuoja iškarto kelis veiksnius (dažniausiai tai yra temperatūra, slėgis ir drėgmė).

1.3. Jutiklių diegimas į avilius

Beveik visi jutikliai, išskyrus infraraudonųjų spindulių jutiklius ir jutiklius, skirtus stebėti išorinius veiksnius, reikalauja, kad jie būtų įdiegti avilio viduje, kas reikalauja papildomo planavimo, kaip juos apsaugoti nuo bičių, Bitės koriams bei savo avilio statybai ir taisymui, apsisaugojimui nuo pašalinių objektų ar kenkėjų (pvz. gyvačių ar driežų) naudoja labai lipnią medžiagą – pikį (angl. *„Propolis“*), todėl bitininkui reikia pasirūpinti elektronikos ir jutiklių apsauga nuo šios medžiagos[10].

Vienas iš būdų apsisaugoti yra naudoti metalinius tinklelius, tvirtinant juos į tarprėmius arba ant avilio sienų iš vidaus, taip padarant savotišką narvelį, kur galima patalpinti jutiklius arba papildomą elektroniką. Jutiklių talpinimas į tarprėmius su tokiais narveliais suteikia galimybę juos išiminėti ir lengvai keisti.

Kitas būdas yra apvynioti jutiklius tam tikru plastiku – teflono (angl. „*PTFE fiber or Polytetrafluoroethylene „*) medžiaga, kuri visiškai apsaugo nuo vandens, drėgmės ar vandens turinčių medžiagų[11]. Šitą medžiagą patartina naudoti ant analoginių jutiklių, kadangi jie neturi jokios apsaugos ant laidų ir pačio jutiklio, kaip tuo tarpu skaitmeniniai jutikliai (pvz. DS18B20) yra atsparūs drėgmei ir turintys gerą izoliaciją ant laidų.

2. Arduino ir maketavimo plokštės apžvalga

Šiame skyriuje yra apžvelgti įrankiai ir prietaisai, kurie skirti prijungti elektroninius jutiklius į grandinę, bei su jais užmegzti ryšį.

2.1. Arduino plokštė

Arduinas yra atviro kodo sistema, skirta programuoti elektroninius prietaisus, kaip pavyzdžiui jutiklius. Pats Arduinas susidaro iš dviejų pagrindinių dalių – fizinės plokštės su mikrovaldikliais ir programinės įrangos, per kurią programuojame ir įkeliame kodą į pačią plokštę. Šitame darbe yra aprašoma ir naudojama Arduino *UNO* plokštė.

Pagrindinės plokštės dalys yra:

- Maitinimas (*USB* jungtis), skirtas prijungti plokštę prie maitinimo šaltinio. Taip pat per šį laidą yra užmezgamas ryšys su kompiuteriu ir keliamas kodas iš kompiuterio į pačią Arduino plokštę.
- Kaištukai (angl. „*Pins“*), skirti jungti laidus, kad sukurti grandinę tarp plokštės ir jutiklių. Kiekvienas plokštės kaištukas turi savo pagrindinę paskirtį, kam jis bus naudojamas. Šiame bakalauro darbe yra naudojami šie kaištukai:
    a. ***GND*** – įžeminimo kaištukas, skirtas įžeminti grandinę.
    b. ***5V/3V*** – įtampos kaištukai, skirti suteikti maitinimą jutikliams.
    c. ***DIGITAL/PVM*** – skaitmeninės įvesties/išvesties kaištukai, skirti komunikacijai su prijungtais jutikliais, bei rinkti skaitmeninius duomenis iš jų.
- LED indikatoriai, kurie teikia tam tikrą informaciją apie pačios plokštės būseną veikimo metu (pvz. ar plokštė yra prijungta prie maitinimo, ar siunčia/gauna duomenis).

Arduinas, kaip platforma, tapo labai populiari tarp žmonių, kurie neturėjo jokios su elektroniniais prietaisais ar programavimu patirties, kadangi platforma siūlo labai draugišką sąveiką su vartotoju. Pavyzdžiui, kad įkelti naują kodą į plokštę, vartotojui nereikia papildomos įrangos – jam tiesiog užtenka USB laido. Be to, programos yra rašomos supaprastinta C++ programinės kalbos versija, kas palengvina kodo rašymą ir jo skaitymą/supratimą su tuo nesusidūrusiems žmonėms arba šios srities naujokams.

Pats kodas susideda iš dviejų pagrindinių dalių – tuščio tipo funkcijų (angl. „*Void“*) *setup()* ir *loop()* (žr. 3 pav.). *Setup()* funkcija yra kviečiama tik vieną kartą ir tik paleidžiant patį programos eskizą(angl. *Sketch*). Pačioje funkcijoje yra kviečiamos bibliotekos, kurios reikalingos jutikliams bei inicijuojami globalūs kintamieji, kurie bus naudojami kitose funkcijose. *Loop()* funkcijos pagrindinė paskirtis yra sudaryti ciklą (angl. „*loop“*), kad programa galėtų tiesiogiai sąveikauti su plokšte ir jos duomenimis – juos keisti ir atitinkamai reaguoti.

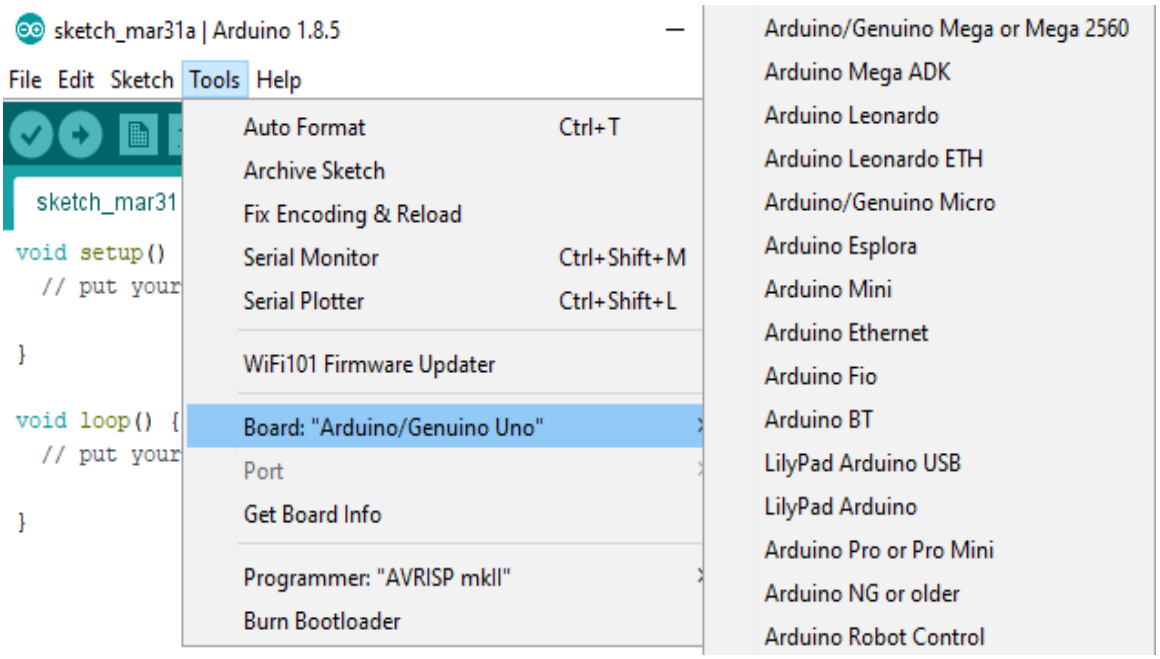
Pati plokštė naudoja nepastovią atmintį(angl. *volatile memory*), todėl išjungus plokštę, visa atmintis yra išvaloma. Tačiau kiekviena Arduino plokštė turi kažkiek **EEPROM** („*Electrically Erasable Programmable Read-Only Memory* “) atminties (512,1024 arba 4096 baitų atminties – dydis priklauso nuo modelio, šiuo atvejų darbe naudojama plokštė turi tik 1024 baitus), kuri yra pastovi ir joje duomenys išlieka visada, net nutraukus maitinimą pačiai plokštei. Naudojantis šia atminti reikia būti atsargiam, kadangi atmintis turi savo gyvavimo ciklą – 100,000 įrašymo/nuskaitymo operacijų. Svarbu suprasti, kad kiekvieno duomens iš baito ištrynimas i įrašymas skaičiuojasi kaip 2 operacijos, o išnaudojus visas baito operacijas – baitas tampa nebenaudojamas.

```
1.  void setup() {
2.
3.  }
4.
5.  void loop() {
6.
7.  }
```

3 pav. Pradinis *Arduino* kodo skeletas

2.2. Programinė įranga „ARDUINO“

„ARDUINO“ yra nemokama, atviro kodo programinė įranga, skirta rašyti kodą elektroniniams prietaisams ir jutikliams ir tą kodą kelti į pačią plokštę. Pati programa yra palaikoma tiek pačių populiariausių operacinių sistemų(*Windows, MAC OS X, Linux*) ,tiek yra sukurtas ir internetinis jos variantas „ARDUINO WEB EDITOR“. Programos privalumas yra tame, jog ji pati palaiko visų tipų Arduino plokštes ir kitus mikrokontrolerius– užtenka tiesiog pasirinkti savo norimą plokštės tipą ar mikrokontrolerį ir programa pati susikonfigūruos pagal ją (žr. 4 pav.).



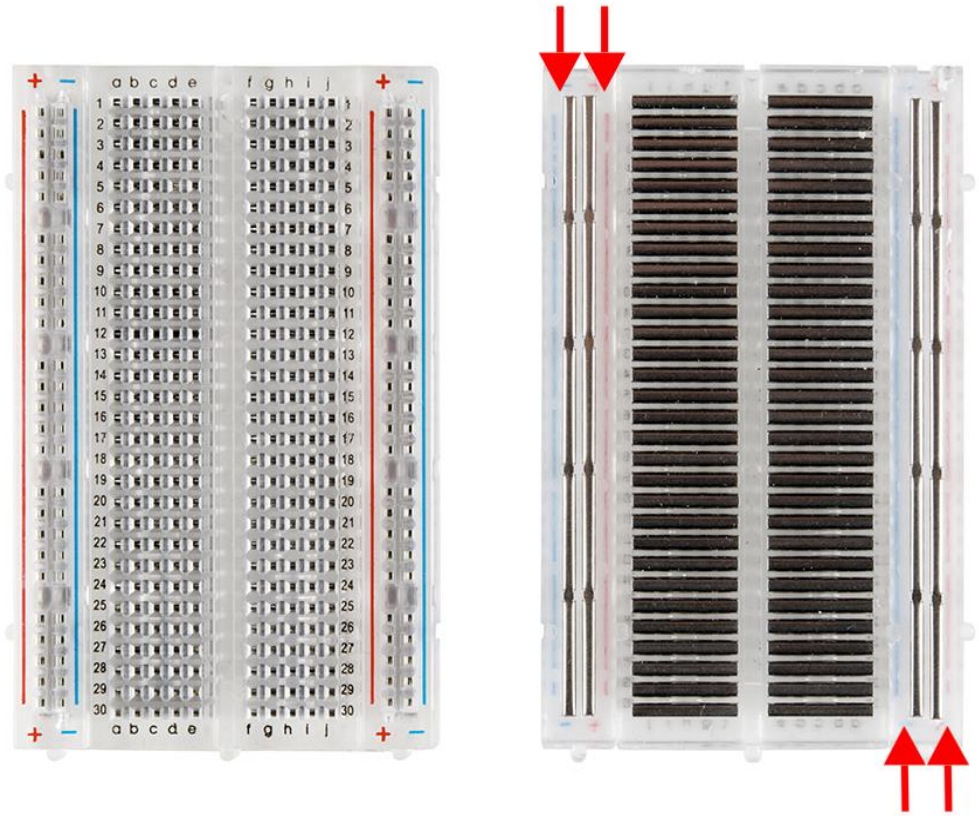
4 pav. Arduino plokštės pasirinkimas programoje

2.3. Maketavimo plokštė

Maketavimo plokštė (angl. "*Breadboard"*) yra skirta kurti laidų grandinėms, kurios jungia laidus, ateinančius iš mikrovaldiklių plokštės į jutiklius. Pagrindinis plokštės privalumas yra tame, jog nereikia nieko lituoti – užtenka laido galus įkišti į tam skirtus kaištukus, kad būtų tiekiama įtampa ten, kur norima. Dėl savo lengvo naudojimo, maketavimo plokštės yra populiarios edukaciniuose tiksluose, kai norima ištestuoti ir parodyti kaip veikia tam tikri elektroniniai prietaisai, jutikliai, ar išanalizuoti prietaisų grandines.

Pagrindinės maketavimo plokštės sudedamosios dalys yra terminalų juostelės (angl. „*Terminal strips"*), per kurias teka srovė. Kiekvienoje eilėje yra po 14 kaištukų, iš kurių 10 yra skirti įjungti komponentus ir 4 kaištukai, per kuriuos teka srovė (angl. „*Power Rails"*) (žr. 5 pav.). Pagrindinis skirtumas tarp juostelių, skirtų jungti komponentus ir įtampos juostelių yra tame, jog visa įtampos juosta yra sujungta viena terminalo juostele, tuo tarpu komponentų juostos yra sujungtos po 5 kaištukus, todėl tiekiant įtampą vienai juostai, kitos juostos įtampos nebeturės.

Didesni projektai, arba projektai, kurie susideda vien tik iš grandies diagramų, reikalauja kelių tokių plokščių, tačiau šiam bakalauro darbui užteko ir vienos.



5 pav. Maketavimo plokštė. Kairėje pusėje yra neišrinktos plokštės vaizdas, dešinėje – išrinktos plokštės vaizdas. Raudona linija žymi teigiamos įtampos juostą, mėlyna žymi neigiamos įtampos juosta, raidės ir skaičiai atitinkamai žymi juostas ir stulpelius.[1]

[1] Piešinukas buvo adaptuotas iš https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/all

3. Temperatūros jutikliai

Projektuose dažniausiai yra naudojami keli temperatūros matavimo jutiklių tipai: analoginiai (angl. „*Analog*“), skaitmeniniai (*angl. „Digital*“) ir rečiau sutinkami - infraraudonieji jutikliai. Tiek analoginiai, tiek skaitmeniniai jutikliai dažniausiai yra termistoriai (angl. „*Thermistors*“), kurie naudoja jautrius temperatūrai rezistorius, kad išmatuoti aplinkos temperatūrą – keičiantis temperatūrai, keičiasi ir rezistoriaus varža[12]. Skaitmeniniai jutikliai dažniausiai susideda iš trijų pagrindinių dalių – jutiklio, laido ir siųstuvo (angl. „*Transmitter*“). Pagrindinis skaitmeninių jutiklių skirtumas nuo analoginių jutiklių yra duomenų perdavimas: skaitmeniniuose jutikliuose duomenys į skaitmeninį signalą yra konvertuojami pačiame jutiklyje, kai tuo tarpu analoginiai jutikliai siunčia analoginius signalus tiesiai į mikrovaldiklį, kur vėliau jie yra apdorojami.

Temperatūrai matuoti taip pat yra naudojama infraraudonųjų spindulių technologija. Infraraudonųjų spindulių prietaisai yra dviejų tipų: skleidžiantys infraraudonuosius spindulius ir spindulius aptinkantis. Temperatūros matavimo atveju, mums svarbūs yra prietaisai, kurie infraraudonuosius spindulius aptinka.

3.1. Temperatūros stebėjimas infraraudonaisiais jutiklis

Infraraudonųjų spindulių (dar vadinami sutrumpintai „IR“) jutikliai veikia pagal sekantį principą – jie mato objekto skleidžiamą energiją (šilumą) ir jeigu objekto temperatūra yra didesnė už absoliutųjį nulį (−273.15° C) - konvertuoja ją į elektroninį signalą, kurį nuskaito jutiklio vaizdo monitorius (*angl. „Photodetector*“). Didžiausias IR pranašumas prieš analoginius termistorius ir skaitmeninius jutiklius yra tas, kad jie veikia per nuotolį. Bitės turi itin jautrią uoslę, todėl visus svetimkūnius avilyje jos yra linkusios užpikiuoti. Tai reiškia, kad naudojant IR jutiklius, bitininkui nereikia papildomai projektuoti jutiklių išsidėstymo avilio viduje, kad gauti tikslią temperatūrą ir tuo pačių netrukdyti bitėms - užtenka pasirinkti atitinkamą atstumą jutikliui, kuris priklauso, nuo to, kokį bangos ilgį (*angl. „wavelength*“) skaito jutiklis. Bangos būna skirtingų tipų: trumpos (14000–4000 cm^{-1} (0.7–2.5 μm bangos ilgis)), vidutinės (4000–400 cm^{-1} (2.5–25 μm bangos ilgis)) arba ilgos (400–10 cm^{-1} (25–1000 μm bangos ilgis)). Norint išgauti tikslų temperatūros paveiksliuką, geriausia jutiklius diegti iš tos pusės, kurioje yra išsidėsčiusi bičių šeima[13].

Tačiau yra ir tam tikri trūkumai, su kuriais yra susiduriama naudojant IR jutiklius. Visu pirma, tai aplinkos veiksniai. Kadangi jutikliai įdiegti avilio išorėje - tiek jutiklis, tiek stebimas objektas (šiuo atveju yra bičių avilys) yra tiesiogiai veikiami išorės veiksnių: lietaus, sniego, vėjo. Norint, kad būtų išgaunama kuo tikslesnė temperatūra, reikia skirti atitinkamą priežiūrą jutikliui, kad jam aplinkos veiksniai darytų kuo mažesnę įtaką, nes nuo to priklauso, kaip tiksliai jutiklis gebės rinkti informaciją apie avilio temperatūrą. Kitas svarbus trūkumas yra kaina. Pigiausi jutiklių variantai turi labai mažą rezoliuciją, todėl jie nematuoja viso objekto šilumos, bet ima kelis taškus ir atitinkamai užpildo plotą, sudarytą iš taškų, parodydama vidutinę dviejų paimtų taškų temperatūrą. Galingesni jutikliai, kurių rezoliucija yra didesnė, gali stebėti objektą iš toliau su didesniu tikslumu, kas leidžia stebėtojui lengviau matyti temperatūros pakitimus objekte – o tokie jutikliai atitinkamai kainuoja kelis kartus brangiau [14].

3.2. Temperatūros matavimas analoginiu termistoriu

Termistoriai yra kintamieji rezistoriai, kurie atitinkamai nuo temperatūros keičia savo varžą. Jie yra analoginiai įrenginiai, todėl jų kodavimas, palyginus su skaitmeniniais jutikliais, yra ganėtinai paprastesnis, nes nereikalauja papildomų bibliotekų, užtenka pasinaudoti keliomis matematinėmis formulėmis, kurios yra aptartos tolimesnėje šio poskyrio dalyje. Kadangi jie veikia kaip rezistoriai, todėl yra naudojami du jų tipai:

Neigiamos temperatūros koeficiento (*angl. „Negative Temperature Coefficient or NTC“*) termistoriai – tai termistoriai, kurių varža mažėja kylant aplinkos temperatūrai.

Teigiamos temperatūros koeficiento (*angl. „Positive Temperature Coefficient or PTC“*) termistoriai – tai termistoriai, kurių varža didėja kylant aplinkos temperatūrai.

Analoginiai termistoriai yra populiarūs dėl savo galimybės matuoti labai žemas ir aukštas temperatūras (dažniausiai jų matavimo diapazonas yra tarp -55°C iki 150°C), bei pasižymi dideliu tikslumu (turi +/-0.2°C tikslumą).

Kadangi termistorius yra kintamasis rezistorius, tai jis leidžia pro save srovę, tuo tarpu Arduino plokštė srovės nemato, o tik ateinančią įtampą, todėl norint sužinoti matuojamą temperatūrą, pradžiai yra apskaičiuojama jutiklio varža. Tam yra naudojama įtampos daliklio (*angl. „Voltage divider“*) lygtis (3.2.1):

$$V_{out} = V_{in} \times (\frac{R2}{R1+R2}) \quad (3.2.1)$$

$$V_{out} - įtampa\ tarp\ termistoriaus\ ir\ žinomo\ rezistoriaus$$

$$V_{in} - šaltinio\ įtampa\ (šiuo\ atveju\ 3{,}5V\ arba\ 5V)$$

$$R2 - termistoriaus\ varža$$

$$R1 - žinomo\ rezistoriaus\ varža$$

Įtampos daliklio formulę galime atitinkamai perdaryti (3.2.2), kad sužinoti naudojamo termistoriaus varžą (dažnai varža iškarto būna nurodyta pas tiekėjus, todėl formulė yra naudojama tik tuo atveju, kai varža yra nenurodyta):

$$R2 = R1 \times (\frac{V_{in}}{V_{out}} - 1) \quad (3.2.2)$$

Paskutinis etapas yra varžos konvertavimas į temperatūrą – tam yra pasitelkiama Steinharto-Harto lygtis (*angl. „Steinhart-Hart Equalation“*) (3.2.3):

$$\frac{1}{T} = A + Bln(R) + C[\ln(R)]^3 \quad (3.2.3)$$

$$T - temperatūra\ Kelvinais$$

$$R - varža\ temistoriuje$$

$$A, B, C - Steinharto - Harto\ koeficientai$$

Termistoriaus tiekėjai dažniausiai patys nurodo *Steinharto – Harto* koeficientus, tačiau kartais tai yra nepadaroma ir tada reikia patiems juos apsiskaičiuoti pagal formulę (3.2.4), kuriai reikia paduoti tris temperatūros jutiklio nuskaitytas temperatūras ir termistoriaus varžą, kad apskaičiuoti nežinomus (A, B, C) kintamuosius:

$$\begin{bmatrix} 1 & \ln(R_1) & \ln(R_1)^3 \\ 1 & \ln(R_1) & \ln(R_1)^3 \\ 1 & \ln(R_1) & \ln(R_1)^3 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \frac{1}{T_1} \\ \frac{1}{T_2} \\ \frac{1}{T_3} \end{bmatrix} \quad (3.2.4)$$

Kad išvengti sunkesnių matematinių skaičiavimų ir supaprastinti rašomą kodą, galima pasinaudoti supaprastinta formule, kuri vadinasi B-parametro lygybė (*angl. „B-parameter equation“*) (3.2.5), į kurią užtenka įsistatyti tam tikras nominalias reikšmes, bei įterpti naudojama varžą **R**, kad išgauti ieškomą temperatūrą **T** Kelvinais. Temperatūra Kelvinais vėliau galima konvertuoti į Celsijus, iš gauto rezultato atėmus **-273.15**:

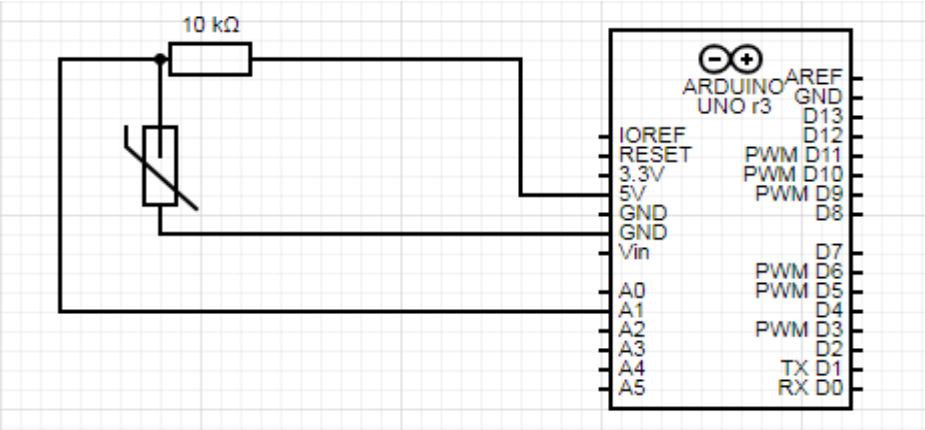
$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B}\ln(\frac{R}{R_0}) \quad (3.2.5)$$

$$T_0 - nominali\ termistoriaus\ temperatūra\ (25℃)\ Kelvinais\ (25℃ = 298.15\ K)$$

$$B - termistoriaus\ koeficientas\ (dažniausiai\ naudojama\ konstanta\ B = 3950)$$

$$R_0 - nominali\ termistoriaus\ varža\ prie\ nominalios\ temperatūros\ (25℃)$$

Apsirašius nurodytas formules, reikia atitinkamai prisijungti termistorių prie mikrovaldiklių plokštės(žr. 6 pav.).



6 pav. Termistoriaus pajungimo prie Arduino plokštės schema[2]

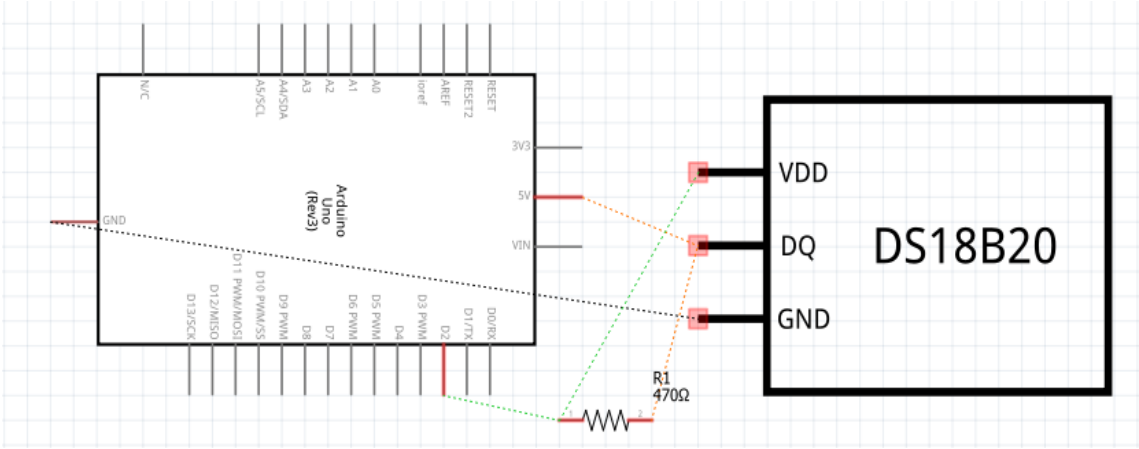
3.3. Temperatūros matavimas skaitmeniniu DS18B20 jutikliu

Skirtingai nei analoginiai termistoriai, skaitmeniniai jutikliai patys atlieka varžos konvertavimą į skaitmeninę įtampą, kuri vėliau siunčiama tiesiai į mikrovaldiklių plokštę. Jų veikimo diapazonas yra šiek tiek mažesnis bei turi didesnę paklaidą (matuoja nuo -55°C iki 125°C su +/-0.5°C paklaida). Pasitelkus kelias bibliotekas („*One-Wire*“ ir

[2] Schema buvo nubraižyta su *Fritzing* programa

„*DallasTemperature*“) yra išvengiamas matematinių formulių kodavimas, kadangi bibliotekos pačios atlieka visus skaičiavimus. Taip pat bibliotekos leidžia rinkti kelių skaitmeninių jutiklių duomenis iš vieno skaitmeninio kaištuko. Bitininkams yra naudinga matuoti avilio temperatūrą skirtingose avilio vietose, todėl dažnai yra poreikis naudoti iš karto kelis jutiklius (dažniausiai naudojami 4 jutikliai, pritvirtinti prie skirtingų avilio sienų, pasitelkus aprašytus patarimus 1.3. poskyryje).

Norint pradėti matuoti temperatūrą, reikia atitinkamai prijungti jutiklius prie mikrovaldiklių plokštės (žr. 7 pav.) ir parašyti kodą, kuriame yra kviečiamos reikalingos bibliotekos (A priedas).



7 pav. Skaitmeninio temperatūros jutiklio pajungimo schema.[3]

Su „OneWire“ biblioteka yra užmezgamas ryšis tarp mikrovaldiklių plokštės ir temperatūros jutiklio. Užmezgus ryšį, plokštė gali pradėti rinkti duomenis iš jutiklio, pasitelkus „DallasTemperature“ biblioteka - pasinaudojant „getTempCByIndex()“ funkcija, temperatūros duomenis yra išvedami aplikacijos konsolėje.

3.4. Temperatūros jutiklių tarpusavio apibendrintas palyginimas

Aprašius temperatūros stebėjimo jutiklius, jų veikimo principus, galima pastebėti, kad kiekvienas iš jų turi atitinkamai savo privalumus ir trūkumus (1 lentelė), kai yra planuojama su jais stebėti bičių avilius.

1 Lentelė – Analoginių, skaitmeninių ir infraraudonųjų spindulių temperatūros jutiklių palyginimas

Jutiklio tipas:	Analoginis	Skaitmeninis	Infraraudonųjų spindulių
Privalumai:	Kodas susideda iš kelių matematinių formulių	„OneWire“ bibliotekos pagalba galima pajungti kelis jutiklius prie vieno duomenų kaištuko	Nereikia jutiklių diegti į avilio vidų

[3] Schema buvo nubraižyta su *Fritzing* programa

	Veikia auštame temperatūros diapazone(-55°C iki 150°C)	„DallasTemperature“ biblioteka apskaičiuoja visas matematines formules	Atvaizduoja temperatūrą ekrane su spalvų gama
	Pasižymi aukštu temperatūros matavimo tikslumu(+/-0.2°C)	Duomenis yra skaitomi srautu per skaitmeninius kaištukus	Tiksliai matoma, kokioje avilio dalyje kokia yra temperatūra
Trūkumai:	Labai jautrus „triukšmams“	Kodas yra kompleksiškesnis nei analoginių daviklių	Jutikliams daro įtaka išoriniai gamtos veiksniai
	Dažnai įvyksta nenuspėjami gedimai	Veikimo diapazonas yra mažesnis nei analoginių daviklių (-55°C iki 125°C)	Už kokybę reikia mokėti labai brangiai
	Jutiklis išveda tik varžą, kurią reikia konvertuoti į įtampą	Palyginus turi aukštą temperatūros paklaidą(+/-0.5°C)	

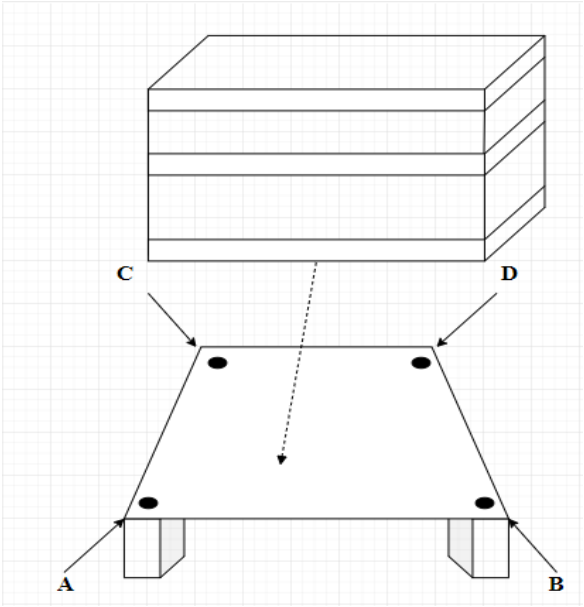
4. Svorio Jutiklis

Arduino svorio matavimo projektuose dažniausiai yra naudojami mažos matavimo galios (~5kg galios) svorio jutikliai. Avilio matavimui tokios galios netinka (4 tokių jutiklių galia yra ~20kg, tuo tarpu aviliai su bitėmis vidutiniškai sveria 80kg), todėl visas dėmėsys bus skiriamas galingesniems, HX711 50kg stiprumo apkrovos jutikliams.

4.1. Hx711 apkrovos jutiklis

HX711 apkrovos jutikliai yra labiausiai paplitę svorio jutikliai mūsų naudojamose įrenginiuose – jie dažniausiai yra naudojami išmaniosiose elektroninėse svarstyklėse. Šie jutikliai dažniausiai būna dviejų tipų: suspaudimo (angl. „*Compression*“) tipo – jie yra dedami po matuojamu objektu, ir įtampos (angl. „*Tension*“), kurie yra išdėstomi atitinkamai, kur norima matuoti objekto keliamą įtampą. Kiekvienas jutiklis turi atskirą įtampos matuoklį (angl. „*Strain gauge*“), kuris deformuojasi, kai jį spaudžia padėto objekto masė. Ta deformacija iškarto yra perduodama kaip kitimas elektros signaluose (varžos kitimas), pagal kuriuos yra apskaičiuojama objekto masė. Jų svorio galia, priklausomai kokio galingumo, svyruoja nuo 1 kilogramo iki 1000 tonų galios (šio darbo dėmėsis yra sutelktas į keturis 50 kilogramų galios jutiklius, kurie leidžia matuoti svorį iki 200 kilogramų). Šitie jutikliai yra populiarūs dėl šių priežasčių:

- Aukštas tikslumas matuojant svorį;
- Ilga veikimo garantija;
- Lengvai įdiegiama;
- Nesunkiai kalibruojama pagal skirtingus standartus (ar tai būtų į kilogramus, ar į svarus).



8 pav. Svorio jutiklių (A, B, C, D) išsidėstymas po bičių aviliu.[4]

[4] Schema buvo nubraižyta su *Draw.io* internetine programa.

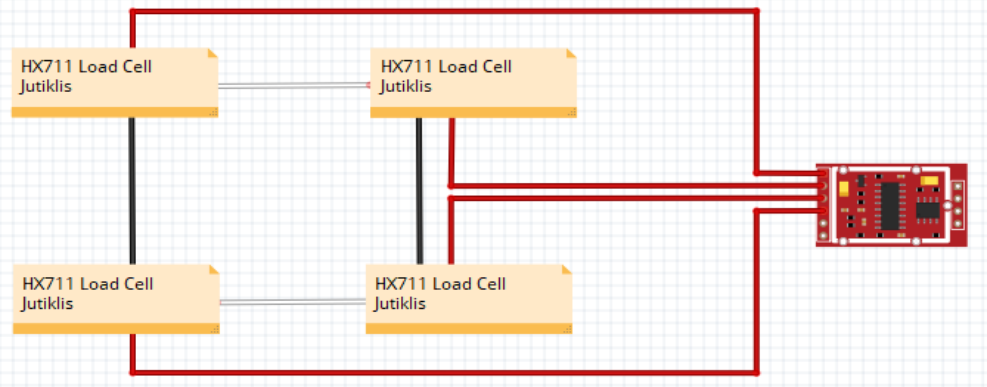
Kiekvienas jutiklis, priklausomai nuo jo maksimalios galios, turi būti kalibruojamas atitinkamai, todėl protingiausia matavimui naudoti vienodus jutiklius. Bičių avilio svorio matavimui yra geriausia naudoti 4 vienodus jutiklius, kad svoris ant jutiklių būtų išsidėstęs vienodai ir svoris būtų išgautas kuo tikslesnis (žr. 8 pav.). Kad Arduino plokštė pajėgtų apdoroti gaunamus duomenis iš 4 prijungtų jutiklių, yra būtina svarstykles jungti prie stiprintuvo (angl. „*Amplifier*“) ir tik tada prie mikrovaldiklių plokštės.

4.2. Hx711 stiprintuvas

Arduino plokštė pati nesugeba nuskaityti svorio jutiklių srovės (varžos) pakeitimų – tam į pagalbą yra naudojamas svorio jutiklių stiprintuvas. Svorio stiprintuvo užduotis yra nuskaityti varžos pakeitimus naudojamose jutikliuose, bei juos konvertuoti į skaitmeninę įtampą. Tuo metu, teisingai sukalibravus svarstykles, yra išgaunamas tikslus matuojamo objekto svoris. Pats stiprintuvas naudoja dviejų laidų sąsają komunikacijai su jutikliais ir plokšte – duomenų ir laiko, o pačios svarstyklės yra jungiamos kietojo tilto jungtimi (angl. *„Wheatstone bridge“*) (žr. 9 pav.), kad varža grandinėje susivienodintų ir svorio stiprintuvas galėtų dirbti su ja. Jutikliai kietojo tilto grandinėje tarpusavyje jungiami sekančia tvarka:

- Kairėje esantys jutikliai sujungiami juodu laidu.
- Viršuje esantys jutikliai sujungiami baltu laidu.
- Dešinėje esantys jutikliai sujungiami juodu laidu.
- Apačioje esantys jutikliai sujungiami baltu laidu.

Atitinkamai raudoni laidai jungiami prie svarstyklių stiprintuvo (į E+, E-, A+ ir A- jungtis), kuris yra jungiamas prie Arduino plokštės (žr. 10 pav.).



9 pav. Kietojo tilto jungtis su 4 svorio jutikliais, kurie prijungti prie HX711 stiprintuvo, leidžiantis matuoti iki 200 kilogramų svorį.[5]

Pati varža kietojo tilto jungtyje apskaičiuojama pagal formulę (4.2.1):

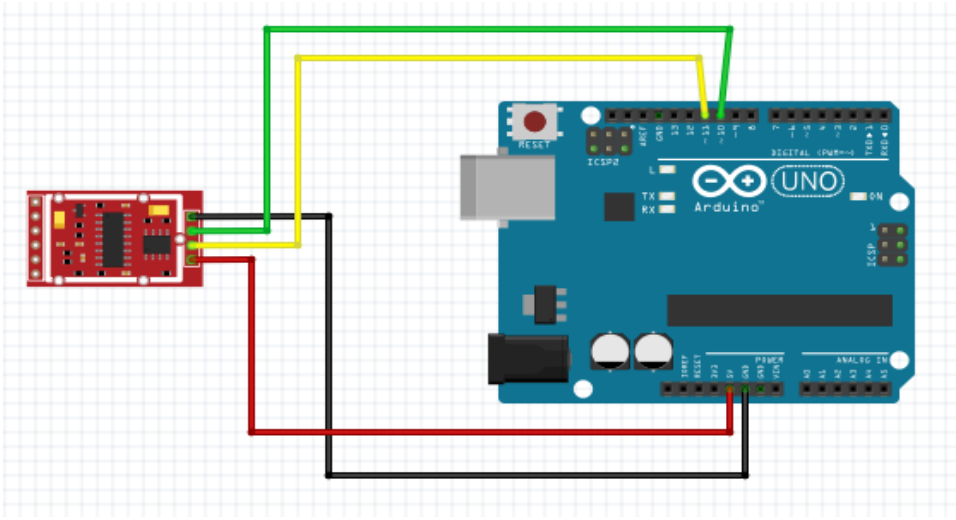
$$\frac{R_2}{R_1} = \frac{R_x}{R_3} \qquad (4.2.1)$$

[5] Schema buvo nubraižyta su *Fritzing* programa

Atitinkamai galime perdaryti lygtį, kad būtų išgautas nežinomas varžos kintamasis(4.2.2):

$$R_x = \frac{R_2}{R_1} \times R_3 \qquad (4.2.2)$$

Formulė padeda surasti nežinomą varžą ir atitinkamai ją sumažinti arba padidinti, kad viso tilto varža būtų tolygiai išdėstyta prieš stiprintuvui ją nuskaitant. Stiprintuvui yra labai svarbu, jog visos varžos būtų suvienodintos, kitaip gali įvykti išgautų duomenų anomalijos.



10 pav. Stiprintuvo jungimas prie Arduino plokštės.[6]

4.3. Svarstyklių kalibravimas ir svorio matavimas

Norint, kad Arduino plokštė leistų kalibruoti ir rinkti duomenis iš svorio jutiklių, reikia pasinaudoti atitinkama svorio jutiklių biblioteka „*HX711*.h“. Kad išgauti tikslius duomenis, svarstykles būtina kalibruoti (Priedas B). Kalibravimas yra padaromas, pasinaudojus bibliotekos „*HX711.h*” teikiama funkcija – “*set_scale()*”, bei nusistačius pradinį kalibravimo koeficientą „*calibration_factor*“.

Norint sužinoti nežinomą koeficientą svarstyklių kalibravimui, užtenka išspręsti tiesinę lygtį (6.3.1):

$$y = m \times x + b \qquad (4.3.1)$$

$$y - \text{žinomas padėtas svoris ant svarstyklių}$$

$$x - \text{gauta reikšmė iš } scale.read_average()$$

$$m - \text{ieškomas nežinomas koeficientas}$$

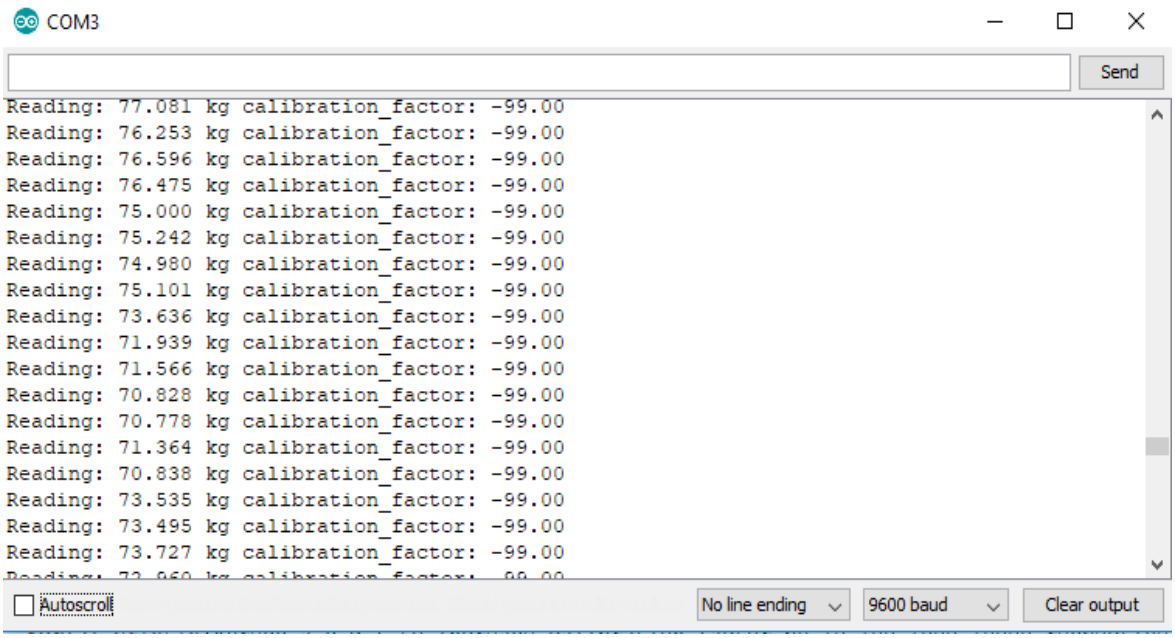
$$b - \text{gauta reikšmė iš svarstyklių be jokių svorių}(scale.tare())$$

Svarbiausia, kad kalibruojant svarstykles, būtų naudojamas atitinkamai didelis svoris (nemažiau 10% - 20% maksimalios svarstyklių galios), nes matuojant su mažais

[6] Schema buvo nubraižyta su *Fritzing* programa

svoriais(pavyzdžiui 1% - 2% maksimalios svarstyklių galios), svarstyklės bus blogai sukalibruotos ir prie didesnių svorių rodys netaisiklingus duomenis.

Taip pat yra tikimybė, kad gerai sukalibravus svorio jutiklius, svoris vis tiek bus rodomas neteisingas. Kaip pavyzdžiui, buvo bandyta kalibruoti senesnes svarstykles prie 72 kilogramų (žr. 11 pav.), keitinėjant kalibravimo faktorių į teigiamą ir neigiamą pusę, tačiau vis tiek buvo gaunama paklaida iki 5 kilogramų.



11 pav. Svorio kalibravimas prie 72 kilogramų.

Susidariusią paklaidą galėjo sukelti viena iš priežasčių:

- Įtampos srovė tekanti iš svorio jutiklių nenuosekliai, todėl reikia tikrinti ar laidais teka pakankamai srovės ir ar ji yra pastovi (tai galima patikrinti su multimetru).
- Svarstyklių jutiklių laiduose yra „triukšmas“, todėl reikia patikrinti ar tie laidai yra nepažeisti, ar jie nėra palaidi (gali būti, kad reikia papildomai perlituoti laidus).

Išsprendus svarstyklių paklaidos problemą, tereikia paduoti gautą kalibravimo faktorių į „*set_scale()*“ funkciją ir su „*get_units()*“ funkcija išvesti gautus duomenis (Priedas C).

5. Bičių avilio monitoringo sistemos kūrimas

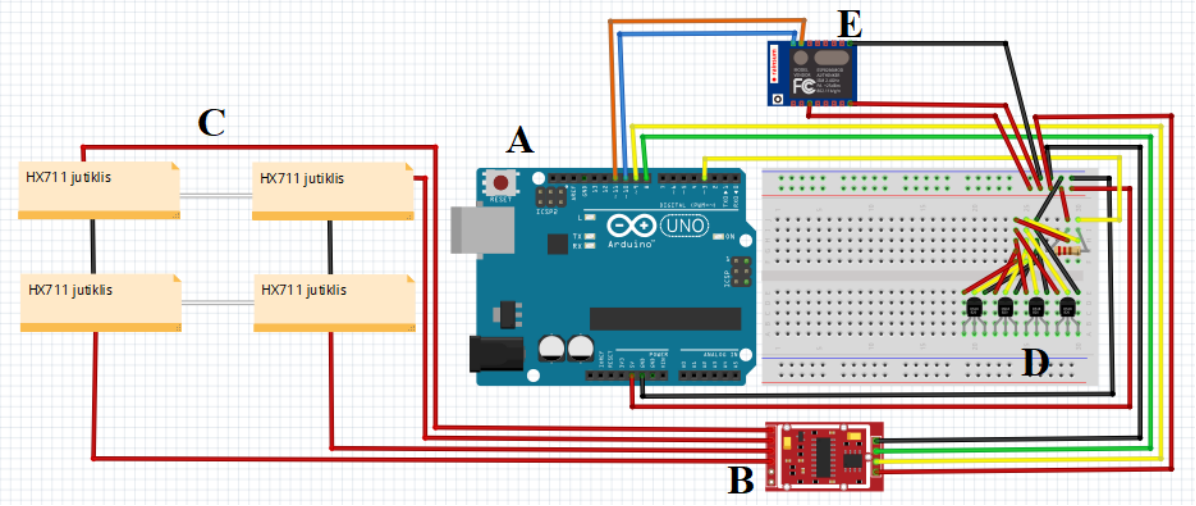
Bičių avilio monitoringo sistemos kūrimas susidaro iš trijų etapų:

- Avilio duomenų rinkimo ir siuntimo;
- Surinktų duomenų saugojimo;
- Saugomų duomenų atvaizdavimo.

Duomenų rinkimui pasitelkta Arduino programinė įranga bei Arduino plokštė. HTTP lokaliam serveriui sukurti naudojama *XAMPP* programinė įranga (*Apache* naudojama *HTTP* serveriui, *MySQL* naudojama duomenų bazei). Duomenų apdorojimas vyksta PHP kalba, o jų atvaizdavimas diagrama yra aprašytas *JavaScript* bei *jQuery* kalbomis. Formų ir navigacijos juostų dizainui pasitelktas *Bootstrap*.

5.1. Avilio duomenų rinkimas ir siuntimas

Duomenims iš jutiklių rinkti yra pasirinkta *Arduino UNO R3* plokštė. Bičių avilio duomenims rinkti, pasirinkti du kintamieji: svoris ir temperatūra. Tiek svoriui, tiek temperatūrai yra naudojami 4 jutikliai – svoriui matuoti naudojami 4 HX711 jutikliai, o temperatūrai 4 DS18B20 jutikliai. Visi 8 jutikliai yra sujungti į vieną grandinę (žr. 12 pav.) ir jų duomenys yra nuskaitomi vienu metu, vieną kartą per *„loop()“* ciklą. Nuskaičius visus duomenis, jie yra siunčiami į serverį, pasitelkus bevielio tinklo modulį ESP8266. Bevielio tinklo modulis su serveriu užmezga ryšį **AT** komandų pagalba (Priedas D).

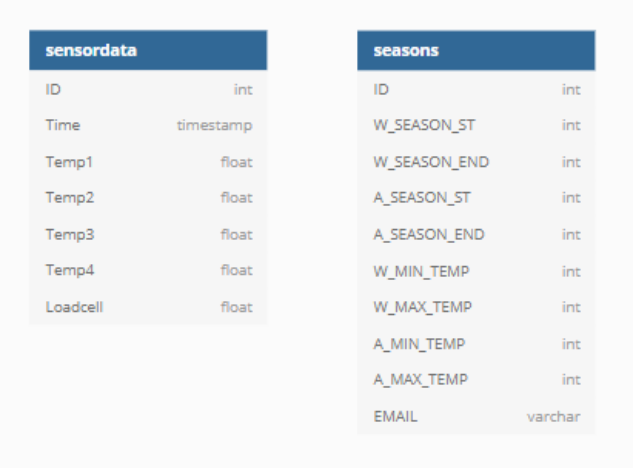


12 pav. Pagrindinė grandinė, skirta stebėti temperatūrą ir masę. „A“ raidė nurodo Arduino UNO R3 plokštę, „B“ raidė nurodo HX711 svarstyklių stiprintuvą, „C“ raidė nurodo HX711 svorio jutiklius, sujungtus į kietąjį tiltą, „D“ raidė nurodo DS18B20 skaitmeninius temperatūros jutiklius, „E“ raidė nurodo ESP8266 bevielio tinklo modulį. Raudonais laidais eina įtampą, juodais laidais eina įžeminimas, o spalvotais laidais(baltais, mėlynais, oranžiniais, geltonais ir žaliu) yra renkami duomenys.

5.2. Surinktų duomenų saugojimas

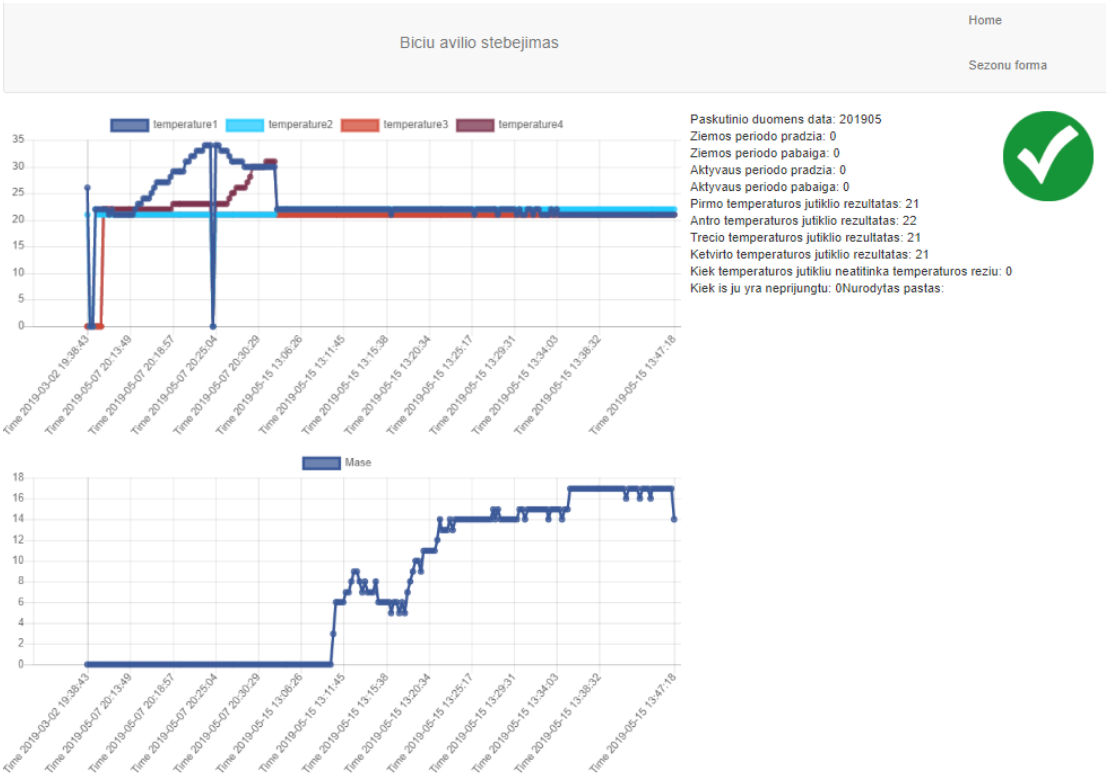
Renkami duomenys yra siunčiami į serverį, kur pasitelkus SQL užklausas (angl. *„queries“*) saugojami duomenų bazėje. Pati bazė susidaro iš dviejų, viena nuo kitos nepriklausomų, lentelių (žr. 13 pav.). Lentelėje „*sensordata*“ yra saugomi duomenys,

atkeliaujantys iš jutiklių: įrašo numeris, visų keturių temperatūros jutiklių išmatuotos temperatūros, svorio jutiklių išmatuotas svoris ir laiko žyma, kada šie duomenys buvo išmatuoti. Lentelėje „*seasons*“ yra saugomi bitininko pasirinkti kriterijai bičių sezonui: miego (žiemos)/aktyvaus sezonų pradžios ir pabaigos, bei tų sezonų temperatūros ribos (minimalios/maksimalios temperatūros tam tikro sezono metu), bei elektroninis paštas.



13 pav. naudojamos lentelės duomenims saugoti.

5.3. Web aplikacija duomenims atvaizduoti



14 pav. pradinis web aplikacijos langas.

Web aplikacijos pagrindinis tikslas yra atvaizduoti surinktus temperatūros ir svorio duomenis diagramomis. Diagramos yra sukurtos naudojantis Chart.js JavaScript biblioteka, duomenys atkeliauja JSON formatu ir jiems atvaizduoti pasitelkti jQuery ir AJAX metodai.

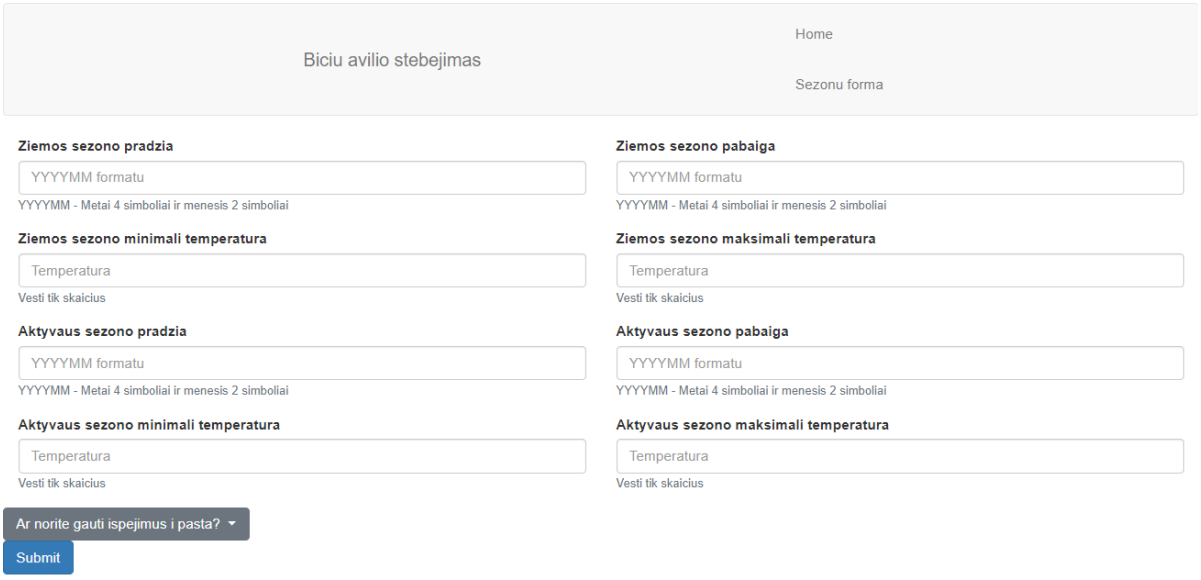
Aplikaciją sudaro 2 langai: Pirmas langas „Home“, skirtas atvaizduoti diagramas ir „Sezonų forma“, skirta užpildyti formą sezonų datoms ir temperatūroms nusistatyti. Pagrindiniame lange (žr. 14 pav.) matomos dvi diagramos – viršutinė skirta stebėti avilio temperatūrai, o apatinė skirta svorio stebėjimui.

Papildomai dešinėje yra nurodomi, kokia yra paskutinio duomens atkeliavimo data (nurodomi tik metai ir mėnesis), kokios yra nusistatytos sezonų datos (pavyzdžio atveju datos yra nenustatytos), bei kokios paskutinės temperatūros buvo nuskaitytos nuo temperatūros jutiklių. Taip pat nurodoma, kiek iš jutiklių nepatenka į bitininko nustatytus rėžius ir kiek iš jutiklių yra neprijungtų. Jutiklių bendrai būsenai yra naudojamos 3 ikonos (žr. 15 pav.): žalia varnelė reiškianti, kad visi temperatūros jutikliai veikia ir atitinka nustatytus rėžius, oranžinis šauktukas, kad vienas jutiklis neatitinka rėžių ir raudonas šauktukas, reiškiantis, kad daugiau nei vieno jutiklio temperatūra neatitinka rėžių ir/arba bent vienas jutiklis yra atsijungęs.



15 pav. bendrai temperatūros būsenai nurodyti panaudoti piešinukai.

Sekančiame lange yra forma (žr. 16 pav.), kurią bitininkas gali užsipildyti, kad galėtų matyti temperatūros jutiklių būsenas tam tikro sezono metu. Papildomai jis gali pridėti savo elektroninį paštą, kad gautų pranešimus į paštą tuo atvejų, jeigu temperatūros jutiklių būsena pasikeičia (pvz. jutiklių matuojama temperatūra nebepatenka į nustatytus rėžius).



16 pav. Sezonų forma skirta nusistatyti datas ir temperatūros rėžius.

Išvados ir rekomendacijos

Pasinaudojant eletriniais jutikliais galima ženkliai sumažinti bitininko darbą, bei išgauti žymei daugiau naudingos informacijos apie patį bičių šeimą, kas ypač yra vertinga žiemą. Matant bičių šeimos informaciją realiu laiku, iškarto yra sumažinamas patikrinimų skaičius, bitininkas visada spės sureaguoti į tam tikrus avilio parametrų pakitimus. Norint pasirinkti teisingus jutiklius, bitininkui rekomenduojama atsižvelgti į šias rekomendacijas:

- Turint didelį biudžetą yra patartina naudoti infraraudonuosius jutiklius, kadangi jie pasižymi savo naudojimo paprastumu, tikslumu ir nereikalauja diegimo į avilio vidų.
- Dėka esamų jutiklių bibliotekų, skaitmeninius daviklius yra paprasčiau naudoti negu termistorius – bibliotekos leidžia atsisakyti matematinės skaičiavimo naštos, o „One-Wire“ biblioteka leidžia vienu kištuku rinkti duomenis iš skirtingų skaitmeninių jutiklių vienu metu.
- Kadangi svarstyklių laidai yra atviri – juos lengva pažeisti, todėl prieš jungiant svarstykles, reikia atidžiai įsitikinti, ar nėra plyšusių arba palaidų laidų, nes tai kelia „triukšmą“, dėl kurio įvyksta didžiulės duomenų paklaidos.
- Jutiklius, kuriuos reikia diegti avilio viduje, geriausia yra apsaugoti, padarant metalinį narvelį. Patartina narvelį tvirtinti prie meduvės arba lizdo rėmelių, nes tai palengvina jutiklių įdiegimą, išėmimą ir priežiūrą.
- Arduino plokštė reikalauja maitinimo per USB kabelį, todėl svarbu apgalvoti, kaip aprūpinti maitinimu sistemą lauke – galima bandyti diegti saulės spindulius renkančias baterijas.

Ateities tyrimų planas

Numatomi ateities darbai yra jau sukurtos sistemos tobulinimas. Reikia papildomai pasirūpinti energijos vartojimo optimizacija – viena iš išeičių būtų pakeisti pagrindinę Arduino plokštę į mažiau energijos vartojantį mikrokontrolerį (pvz. naudojant Atmega328 mikrokontrolerį). Kitas žingsnis yra sugalvoti sprendimo būdą, kaip galima būtų identifikuoti Arduino plokštes viena nuo kitos, kas leistų stebėti iškarto kelis avilius. Taip pat patobulinti WEB aplikaciją, kad ji sugebėtų išvedinėti duomenis iš kelių avilių.

Literatūros sąrašas

[1] Armands Kviesis, Aleksejs Zacepins. *System Architectures for Real-time Bee Colony Temperature Monitoring*, ICTE in Regional Development, 2014, p. 3-7.

[2] Marcus Foth, Alethea Blackler, Paul Cunningham. *A Digital Beehive Could Warn Beekeepers When Their Hives are Under Attack*, 2016.

[3] Paul Allison. *Can a Beehive Get Too Hot or Too Cold?* 2018.

[4] Aston D. *Temperature and thermoregulation in the beehive*, http://www.arnia.co.uk/temperature-and-thermoregulation-in-the-beehive/

[5] Niels Holst, William G. Meikle. *Breakfast Canyon Discovered in Honeybee Hive Weight Curves*. 2018.

[6] Viktor Pankov. *Weight my Bees: Measuring Beehive Weight to Monitor Colony Health.* 2018.

[7] Jonas Kriščiūnas. *Geriausi Aviliai.* 1933, p. 16.

[8] Nurul Hida Zainuddin, Noriah Bidin, Majda Abdullah, M. Yasin *Fiber optic displacement sensor for honey purity detection in distilled water*. 2013.

[9] Chiu Chen, En-Cheng Yang, Joe-Air Jiang, Ta-Te Lin. *An imaging system for monitoring the in-and-out activity of honey bees*. 2012.

[10] Vijay D. Wagh *Propilis: A Wonder Bees Product and Its Pharmacological Potentials.* 2013.

[11] Nathan Seidle. *The Internet of Bees: Adding Sensors to Monitor Hive Health.* 2015.

[12] Brian Gosselin. *NTC Thermistors versus voltage output IC temp sensors*.

[13] Alexander Chiltion. *The Working Principle and Key Applications of Infrared Sensors.* 2014.

[14] Bee Culture The Magazine of American Beekeeping. *Infrared: The next generation in colony management.* 2015.

PRIEDAS A

Kodo fragmentas temperatūros matavimui su DS18B20 jutikliais. Kviečiama „*OneWire*“ bibliotekos nuoroda, kurios pagalba nurodoma, į kurį kaištuką eis duomenys iš skirtingų jutiklių. „*DallasTemperature*“ bibliotekos pagalba išsikviečiame funkcijas „*requestTemperatures()*“ ir „*getTempCByIndex()*“ temperatūrai nuo jutiklių nuskaityti.

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

//Nurodoma, į kurį duomenų kištuką yra prijungtas jutiklis(2 kištukas)
OneWire oneWire(2);

//Paduodama jutiklio nuorodą(instance) į "DallasTemperature" biblioteką
DallasTemperature sensors(&oneWire);

void setup(void){
 Serial.begin(9600);
 Serial.println("Temperatūros matavimas DS18B20 jutikliais");
 sensors.begin();
}

void loop(void){
 //Kvieciama funkcija, kuri paduoda jutikliams komandą, kad pradėtų
 //siusti duomenis
 sensors.requestTemperatures();
 //Renkami duomenis iš keturių prijungtų jutiklių
 for (int i = 0; i <= 4; i++){
    Serial.println(i+1, " jutiklio temperatura: ", sensors.getTempCByIndex(i));
 }
 //Uždedamas delsimas ant duomenų išvedimo
 delay(1000);
}
```

PRIEDAS B

Kodo eskizas, skirtas kalibruoti HX711 svarstykles. Paleidus programą, reikia eiti į konsolę (paspaudus CTRL+SHIFT+M) ir atitinkamai spaudinėti klavišus: „1“, “2“, “3“, “4“, kad padidinti koeficiento kintamąjį ir „7“, „8“, „9“, „0“, kad sumažinti koeficiento reikšmę.

```
#include "HX711.h"

#define DOUT 9
#define CLK 8

HX711 scale;
float calibration_factor = -620; // kalibravimo koeficientas

void setup() {
  Serial.begin(9600);
  scale.begin(DOUT, CLK);
  scale.set_scale(); // įjungiamos svarstyklės
  scale.tare(); // išvaloma svarstyklių atmintis
}

void loop() {
  scale.set_scale(calibration_factor); // paduodame kalibravimo kintamąjį.
  // kad kintamąjį padidinti reikia suvesti į konsolę 1;2;3;4
  // kad sumažiniti: 7;8;9;0
  Serial.print("Nuskaitymas: ");
  Serial.print(scale.get_units(), 1); // išvedamas svoris su dešimtąja dalimi
  Serial.print(" kg");
  Serial.print(" calibration_factor: ");
  Serial.print(calibration_factor);
  Serial.println();
  delay(1000);
  if (Serial.available()) { //sąlyga, kad galima būtų keitinėti kalibravimo faktorių.
    char temp = Serial.read();
    switch (temp) {
    case '1':
      calibration_factor += 1;
      break;
    case '2':
      calibration_factor += 10;
      break;
    case '3':
      calibration_factor += 100;
      break;
    case '4':
      calibration_factor += 1000;
      break;
    case '7':
      calibration_factor -= 1;
      break;
    case '8':
      calibration_factor -= 10;
      break;
    case '9':
      calibration_factor -= 100;
      break;
    case '0':
      calibration_factor -= 1000;
      break;
    }
  }
}
```

PRIEDAS C

Svorio nuskaitymas iš sukalibruotų HX711 svarstyklių. Duomenų nuskaitymui, pasitelkta komanda „*get_units()*“ iš „*HX711.h*“ bibliotekos

```
#include "HX711.h"
#define calibration_factor 99 //gauta reikšmė iš kalibravimo
HX711 scale(11, 10);

void setup() {
  Serial.begin(9600);
  scale.set_scale(calibration_factor); //Paduodame gautą kalibravimo koeficientą
  scale.tare(); //išvalome svarstyklių atmintį
}
void loop() {
  Serial.print("Reading: ");
  Serial.print(scale.get_units(), 1);
  Serial.print(" kg");
  Serial.println();
}
```

PRIEDAS D

Bevielio tinklo modulio naudojamos komandos komunikacijai bei duomenų užsiuntimui. Adaptuota iš https://nurdspace.nl/ESP8266#AT_Commands

Commands	Description	Set/Execute	Parameters
AT+RST	restart the module	–	–
AT+CWMODE	Wi-Fi mode	AT+CWMODE=<mode>	1= Sta, 2= AP, 3=both
AT+CWJAP	join the AP	AT+ CWJAP =<ssid>,< pwd >	ssid = ssid, pwd = wifi password
AT+CWLAP	list the AP	AT+CWLAP	
AT+CWQAP	quit the AP	AT+CWQAP	
AT+ CWSAP	set the parameters of AP	AT+ CWSAP= <ssid>,<pwd>,<chl>, <ecn>	ssid, pwd, chl = channel, ecn = encryption
AT+ CIPSTATUS	get the connection status	AT+ CIPSTATUS	
AT+CIPSTART	set up TCP or UDP connection	1)single connection (+CIPMUX=0) AT+CIPSTART= <type>,<addr>,<port>; 2) multiple connection (+CIPMUX=1) AT+CIPSTART= <id><type>,<addr>, <port>	id = 0-4, type = TCP/UDP, addr = IP address, port= port
AT+CIPSEND	send data	1)single connection(+CIPMUX=0) AT+CIPSEND=<length>; 2) multiple connection (+CIPMUX=1) AT+CIPSEND= <id>,<length>	
AT+CIPCLOSE	close TCP or UDP connection	AT+CIPCLOSE=<id> or AT+CIPCLOSE	
AT+CIFSR	Get IP address	AT+CIFSR	
AT+ CIPMUX	set multiple connection	AT+ CIPMUX=<mode>	0 for single connection 1 for multiple connection
AT+ CIPSERVER	set as server	AT+ CIPSERVER= <mode>[,<port>]	mode 0 to close server mode, mode 1 to open; port = port
+IPD	received data		