

KLAIPĖDOS UNIVERSITETAS
JŪROS TECHNOLOGIJŲ IR GAMTOS MOKSLŲ FAKULTETAS
INFORMATIKOS IR STATISTIKOS KATEDRA

**MIESTO AUTOBUSO KOMPOZITINIO KĖBULO
DEFORMACIJŲ INFORMACINĖS SISTEMOS KŪRIMAS**
**THE DEVELOPMENT OF INFORMATION SYSTEM FOR CITY BUS COMPOSITE
BODY DEFORMATIONS**

Magistro baigiamasis darbas
Techninių informacinių sistemų inžinerijos programa valstybinis kodas 6211BX020

Autorius	JMNTS18 gr. stud. Mantas Tadaras
Vadovas:	Doc. Dr. Mindaugas Kurmis
Konsultantas:	Doc. Dr. Artūras Tadžijėvas

Klaipėda, 2020

MAGISTRO BAIGIAMOJO DARBO LYDRAŠTIS

Pildo magistro baigiamojo darbo autorius

Mantas Tadaras

(bakalauro baigiamojo darbo autoriaus vardas, pavardė)

Baigiamojo darbo tema: Miesto autobuso kompozitinio kėbulo deformacijų informacinės sistemos kūrimas.

The development of information system for city bus composite body deformations.

(magistro baigiamojo darbo pavadinimas lietuvių kalba)

Patvirtinu, kad magistro baigiamasis darbas parašytas savarankiškai, nepažeidžiant kitiems asmenims priklausančių autorių teisių, visas baigiamasis bakalauro darbas ar jo dalis nebuvo panaudotas Klaipėdos universitete ir kitose aukštosiose mokyklose.

Mantas Tadaras

(magistro baigiamojo darbo autoriaus vardas, pavardė ir parašas)

Sutinku, kad magistro baigiamasis darbas būtų naudojamas neatlygintinai 5 m. Klaipėdos universiteto studijų procese.

Mantas Tadaras

(magistro baigiamojo darbo autoriaus vardas, pavardė ir parašas)

Pildo magistro baigiamojo darbo vadovas

Magistro baigiamąjį darbą ginti

(įrašyti – leidžiu arba neleidžiu)

doc. dr. Mindaugas Kurmis

(bakalauro baigiamojo darbo vadovo vardas, pavardė ir parašas)

.....
(data)

Pildo katedros, kuriojančios studijų programą, administratorius (sekretorius)

Baigiamasis darbas įregistruotas katedroje

(data)

Laima Brazdeikienė

(katedros sekretorės vardas, pavardė ir parašas)

Pildo katedros, kuriojančios studijų programą, vedėjas

Magistro baigiamąjį darbą ginti

(įrašyti – leidžiu arba neleidžiu)

doc. dr. Mindaugas Kurmis

(katedros vedėjo vardas, pavardė ir parašas)

.....2020.....
(data)

Recenzentu(-ais) skiriu

.....
(įrašyti recenzento(ų) vardą, pavardę)

doc. dr. Mindaugas Kurmis

(programos vadovas. vardas, pavardė ir parašas)

.....
(data)

SANTRAUKA

Tadaras, Mantas. Miesto autobuso kompozitinio kėbulo deformacijų informacinės sistemos kūrimas. Vadovas doc. dr. Mindaugas Kurmis. Klaipėdos universitetas, Jūros technologijų ir gamtos mokslų fakultetas, Informatikos ir statistikos katedra. Klaipėda, 2020, 53 p.

Kompozitinės medžiagos vis labiau naudojamos įvairių rūšių transporto priemonėse dėl savo savybių lyginant su tradicinėmis medžiagomis. Tačiau kompozitinėse medžiagose nėra aiški jų patvarumo riba. Todėl pasiekusios tam tikrą deformacijų skaičių, kompozitinės medžiagos pasižeidžia, todėl yra svarbu fiksuoti kompozitinėse medžiagose įvykstančias deformacijas, kad išvengti katastrofinių padarinių.

Išnagrinėjus kompozitines medžiagas, jų deformacijos matavimo būdus ir panašias sistemas yra iškeliami reikalavimai ir pasiūlymai naujai kuriamos kompozitinio kėbulo deformacijų informacinei sistemai. Kuriamą sistemą susideda iš dviejų pagrindinių dalių: deformacijų fiksavimo mazgo ir jų išorinės informacinės sistemos.

Testavimo metu prototipas buvo išskaidytas į dvi atskiras dalis: deformacijos mazgo ir išorinės informacinės sistemos testavimą. Deformacijos mazgo testavime buvo atliktas kiekvieno programinės įrangos modulių testavimas. Deformacijos fiksavimo modulio testavimo metu deformacijų fiksavimo duomenys buvo sulyginami su komerciškai prieinamu įtempiu matuokliu Spider8. Deformacijų išsaugojimo modulyje buvo atliktas SD kortelės testavimas, o deformacijos išsiuntimo metu buvo patikrintas duomenų išsiuntimas MQTT protokolu. Išorinės informacinės sistemos testavimo metu buvo patikrinti funkciniai reikalavimai ir atliktas suderinamumo testas su įvairiais įrenginiais ir naršyklėmis. Po to bendras sistemos prototipas yra išbandomas realiomis sąlygomis integravus į elektrinio miesto autobuso „Dancer“ kėbulą.

RAKTINIAI ŽODŽIAI: informacinė sistema, įtempių jutiklis, kompozitinės medžiagos, deformacijų fiksavimas, MQTT.

SUMMARY

Tadaras, Mantas. The development of information system for city bus composite body deformations. Supervisor doc. dr. Mindaugas Kurmis, Klaipeda University. The Faculty of Marine Technology and Natural Sciences. The Department of Informatics and Statistics. Klaipėda, 2020, 53 p.

Composite material are increasingly used in various types of transport due to their properties compared to traditional materials. However, in composite materials there are no endurance limit to their durability. Therefore, after reaching certain number of deformations, the composite materials are damaged. So it is important to record occurring deformations in composite materials to avoid catastrophic consequences.

After analyzing composite materials and their deformations measurement systems, requirement and suggestions for the new composite body deformation system was created. The developed system consists of two parts: deformation measurement device and external information system,

Testing was divided into two separate parts: deformation measurement device and external information system. In the deformation measurement device every software module was tested. During testing of the strain measurement data was compared with commercially available strain gauge Spider8. During deformation strain saving testing, SD card was tested and finally strain data transmission was tested using MQTT protocol. During external information system the functional requirements were checked. Also compalidity test was performed with various devices and browsers. Then prototype was integrated into the body of electric city bus „Dancer“ and tested in real conditions.

KEYWORDS: information system, strain gauge, composite materials, deformation measurement, MQTT.

TURINYS

LENTELIŲ SĄRAŠAS	6
PAVEIKSLĖLIŲ SĄRAŠAS	7
TERMINŲ IR SANTRUMPŲ ŽODYNAS	9
ĮVADAS	10
1. MIESTO AUTOBUSO KOMPOZITINIO KĖBULO DEFORMACIJŲ TEORINĖ DALIS .	11
1.1. Kompozitinių medžiagų apžvalga.....	11
1.2. Kompozitinių medžiagų deformacijų matavimas	12
1.3. Duomenų surinkimo sistemos.....	14
1.4. Įtempių jutiklių duomenų surinkimo sistemų analizė.....	15
1.4.1. Deformacijų matavimas su ir įtempio jutikliais, naudojantis Arduino mikrovaldikliu ..	15
1.4.2. Mažos kainos, didelio greičio, daugiakanalė Arduino paremta duomenų surinkimo sistema vėjo turbinų sistemoms.	17
1.4.3. Bluetooth technologija paremta belaidė įtempių matavimo sistema, krovinio transporto perkrovimui stebėti.....	18
1.5. Išanalizuotų duomenų surinkimo sistemų palyginimas ir siūlomi tobulinimo sprendimai	19
2. MIESTO AUTOBUSO KOMPOZITINIO KĖBULO DEFORMACIJŲ SISTEMOS PROJEKTAVIMAS	21
2.1. Miesto autobuso kompozitinio kėbulo deformacijų sistemos modelis.....	21
2.1.1. Funkciniai ir nefunkciniai reikalavimai.....	23
2.1.2. Sistemos naudotojai	24
2.2. Miesto autobuso kompozitinio kėbulo deformacijų sistemos prototipas.....	24
2.2.1. Deformacijų fiksavimo prototipas	24
2.2.2. Deformacijų informacinė sistema.....	30
3. MIESTO AUTOBUSO KOMPOZITINIO KĖBULO DEFORMACIJŲ SISTEMOS EKSPERIMENTIS TYRIMAS	39
3.1. Deformacijų fiksavimo mazgo testavimas.....	39
3.1.1. Deformacijų fiksavimo modulio testavimas.....	39
3.1.2. Deformacijos fiksavimo išsaugojimo modulio testavimas.....	41
3.1.3. Deformacijos fiksavimo išsiuntimo modulio testavimas.....	41
3.2. Informacinės sistemos testavimas.....	42
3.3. Miesto kompozitinio kėbulo deformacijų fiksavimo prototipo bandymas	44
IŠVADOS	46
LITERATŪROS SĄRAŠAS	48
PRIEDAI	51

LENTELIŲ SĄRAŠAS

1 lentelė.	Gauti rezultatai matuojant deformacijas.....	16
2 lentelė.	Struktūrinės diagramos simbolių paaiškinimas	22
3 lentelė.	Sistemos administratoriaus rolė.....	24
4 lentelė.	Sistemos inžinieriaus rolė.....	24
5 lentelė.	ESP modulių skirtumai.....	26

PAVEIKSLĖLIŲ SĄRAŠAS

1 pav.	Laminatų sustiprinimo būdai: a) ilgi pluoštai b) maži pluoštai c) tekstilės pluoštas d) sluoksnis kitos medžiagos e) išbarstytos dalelės	11
2 pav.	Deformacijos ilgio pokytis.....	12
3 pav.	Deformacijos rūšys: a) ašinė b) lenkimo	12
4 pav.	Įtempio jutiklio sandara	13
5 pav.	Vitstono tiltelis skirtas matuoti mažus pasikeitimus varžoje.....	13
6 pav.	Tipinė duomenų surinkimo sistemos sandara	14
7 pav.	Sistemos jungimo diagrama, skirta duomenų surinkimui.....	16
8 pav.	Įtempių jutiklių priklijavimas prie: a) pirmos sistemos b) antros sistemos	16
9 pav.	Duomenų surinkimo sistemos blokinė diagrama	17
10 pav.	Siūlomos sistemos sandara.....	19
11 pav.	Kuriamos sistemos struktūrinė diagrama	21
12 pav.	Miesto autobuso kompozitinio kėbulo deformacijų fiksavimų sistemos prototipo diegimo diagrama	25
13 pav.	Analoginio skaitmeninio keitiklio modulio išdėstymas	27
14 pav.	SD kortelės modulio išdėstymas	27
15 pav.	Naudojamo įtempių jutiklio struktūra	28
16 pav.	Arduino IDE pagrindinis programos langas.....	28
17 pav.	Deformacijų fiksavimo mazgo prototipo elektroninė schema	29
18 pav.	Surinktas deformacijų fiksavimo mazgo prototipas.....	30
19 pav.	Panaudos atveju diagrama	31
20 pav.	Google Trends paieškų skaičius apie PHP karkasus.....	32
21 pav.	MVC modelio schema.....	33
22 pav.	Sukurtos išorinės deformacijų sistemos klasių diagrama.....	33
23 pav.	Sukurtos išorinės deformacijų sistemos sekų diagrama.....	34
24 pav.	Kuriamos informacinė sistemos diegimo diagrama	35
25 pav.	Išorinės deformacijų informacinės sistemos pradinis langas	36
26 pav.	Išorinės deformacijų informacinės sistemos skydelis a) kompiuteriuose b) planšetėje..	36
27 pav.	Išorinės deformacijų informacinės sistemos išskleistas meniu sąrašas.....	37
28 pav.	Išorinės deformacijų informacinės sistemos vartotojų administravimo langas	37
29 pav.	Išorinės deformacijų informacinės sistemos mazgo pridėjimo/redagavimo langas.....	37
30 pav.	Išorinės deformacijų informacinės sistemos mazgo temų administravimo langas	38
31 pav.	Kompozitinio kėbulo testavimo bandinys.....	39
32 pav.	Kompozitinio kėbulo bandinio testavimas staklėse	40
33 pav.	Deformacijų fiksavimo palyginimo grafikai	40

34 pav.	SD kortelės modulio testavimas.....	41
35 pav.	Išsiųsti duomenys į MQTT brokerį.....	42
36 pav.	Išorinės deformacijų informacinės sistemos skydelio langas, skirtinguose įrenginiuose	43
37 pav.	Sistemos prototipo bandymui buvo pasirinktas elektrinis miesto autobusas „Dancerbus“	44
38 pav.	Bandomojo važiavimo maršrutas	44
39 pav.	Testavimo metu surinktų deformacijų fiksavimo duomenys	45

TERMINŲ IR SANTRUMPŲ ŽODYNAS

ADC - analoginis/skaitmeninis keitiklis (angl. Analog-to-Digital converter).

CSS - pakopiniai stilių šablonai (angl. Cascading Style Sheets).

PK – pirminis raktas. (angl. primary key)

FK – išorinis raktas. (angl. foreign key)

UML– objektinis modeliavimo standartas – vieninga modeliavimo kalba. (angl. Unified Modeling Language)

PHP – dinaminė interpretuojama programavimo kalba (angl. Hypertext Preprocessor)

HTTP – hiperteksto perdavimo protokolas (angl. Hypertext Transfer Protocol)

MVC – modelis-vaizdas-valdiklis šablonas (angl. Model View Controller)

DAQ – duomenų surinkimas(angl. Data acquisition)

SD – atminties kortelė (angl. secure digital)

MQTT – žinučių eilės sudarymo telemetrijos transporto protokolas (angl. Message Queuing Telemetry Transport Protocol)

M2M – veikimo principas: prietaisas prietaisui (angl. machine-to-machine) RAM - adresuojama atmintis (angl. Random Access Memory).

ROM - pastovioji atmintis (angl. Read Only Memory).

SPI - nuosekli periferinė sąsaja (angl. Serial Peripheral Interface).

I2C – tarpusavyje integruotas protokolas (angl. Inter-integrated Circuit Protocol)

UART - universalus asinchroninis imtuvas-siūstuvus (angl. Universal Asynchronous Receiver – Transmitter).

PCB - spausdinta schema (angl. Printed Circuit Board)

TCP / IP – transporto valdymo protokolas / interneto protokolas,(angl. Transport Control Protocol / Internet Protocol)

IoT – Daiktų internetas (angl. Internet of things)

WEB – Saitynas (angl. World Wide Web)

HTML - hiperteksto žymėjimo kalba (angl. Hyper Text Markup Language).

WiFi – belaidis tinklas (angl. wireless fidelity)

ĮVADAS

Kompozitinės medžiagos vis daugiau naudojamos lėktuvų, erdvėlaivių, laivų, automobilių, viešo transporto gamyboje. Tai lėmė jų stiprumas, tvirtumas, lengvumas ir atsparumas korozijai lyginant su tradicinėmis medžiagomis[1, 2, 3]. Tačiau patirdamos apkrovą kompozitinės medžiagos gali įtrūkti ar atsiskuksniuoti, taip sukeldamos katastrofinius padarinius. Kadangi daugelis kompozitinių medžiagų defektų prasideda mikroskopinėje skalėje, todėl yra neįmanoma pamatyti be specialios įrangos[4]. Dėl savo kainos ir paprastumo, deformacijų matavimui yra naudojami įtempių jutikliai[5, 6]. Jie gali būti optiniai, mechaniniai arba elektriniai, kurie paremti varža, talpumu, induktyvumu ir fotoefektu[7, 8]. Dažniausiai yra naudojami elektrinės varžos įtempių jutikliai, kuriuose jos pokytis nusako medžiagos standumą. Daugelis kompozitinių medžiagų deformacijų matavimų su įtempių matuokliais yra atliekama kontroliuojamoje aplinkoje, kurioje naudojamos duomenų rinkimo sistemos[9, 10].

Nors egzistuoja daugybė duomenų surinkimo sistemų, tačiau susiduriama su jų limitais. Vienas iš jų limituotas įvesčių skaičius, dėl to kai norima nuskaityti deformacijas iš daugybės skirtingų taškų, reikia prijungti daugiau duomenų rinkimo įrenginių, dėl ko didėja tokios sistemos kaina ir kompleksškumas. Iškyla problemų, kai norima peržiūrėti nuskaitytas reikšmes, nes jos yra saugomos tiesiogiai į kompiuteryje esančią programinę įrangą. Taip pat tokios sistemos reikalauja pastovios priežiūros, kad veiktų taisyklingai[11].

Darbe bus siūloma miesto autobuso kompozitinio kėbulo ir jo sujungimų, deformacijų fiksavimo informacinė sistema, susidedanti iš mazgų, kuris matuoja deformacijas skirtinguose kompozitinio kėbulo vietose ir serverio, kur duomenys bus saugomi ir atvaizduojami tolimesnei analizei. Duomenys bus perduodami MQTT protokolu, kuris užtikrina lengvą tinklo išplėtimą, sunaudojama nedidelį duomenų srautą, perduodant duomenis [12], dėl to yra tinkamas šiame uždavinyje.

Baigiamo darbo tikslas sukurti realaus laiko autobuso kompozitinio kėbulo deformacijų fiksavimo informacinę sistemą.

Šiam tikslui pasiekti buvo išsikelti **uždaviniai**:

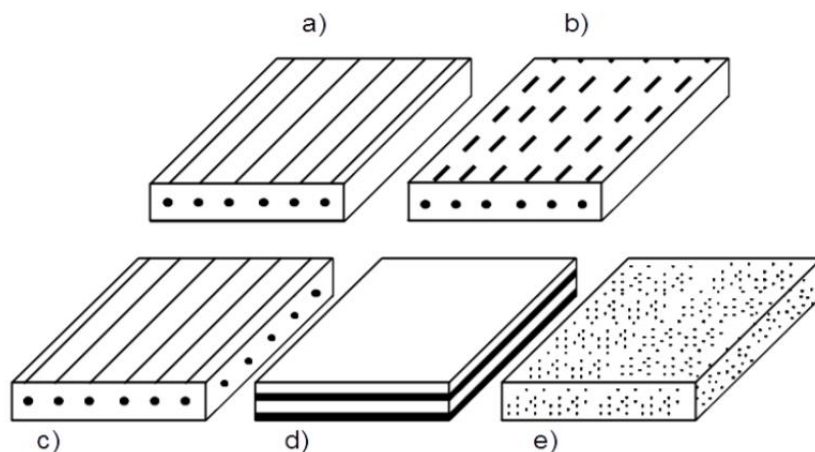
1. Išanalizuoti kompozitinių medžiagų deformacijų fiksavimo būdus ir jų duomenų rinkimo sistemas.
2. Suprojektuoti realaus laiko sistemą, skirtą miesto autobuso kompozitinio kėbulo deformacijų fiksavimui ir atvaizdavimui.
3. Sukurti prototipą, skirtą autobuso kompozitinio kėbulo deformacijų matavimui.
4. Atlikti suprojektuotos sistemos prototipo eksperimentinį tyrimą ir palyginti gautus rezultatus.

1. MIESTO AUTOBUSO KOMPOZITINIO KĖBULO DEFORMACIJŲ TEORINĖ DALIS

1.1. Kompozitinių medžiagų apžvalga

Kompozitai žmonijai žinomi daugiau kaip tūkstančius metų. Jau senovėje ši technologija buvo naudota pagaminti kompozitinę plytą, kaip tvirtą statybos medžiagą. Tačiau kompozitų naudojimas dideliu mastu įvyko neseniai. Dėl tendencijų kurti vis kitokiomis savybėmis pasižyminčias medžiagas, kompozitines medžiagas tapo populiarios tarp konstrukcinių medžiagų. Dėl to kompozitines medžiagas vis dažniau naudojamos kasdieniniame gyvenime ir sutinkamos lėktuvų, erdvėlaivių, laivų, automobilių, viešo transporto industrijose[13].

Kompozitinės medžiagos yra padalintos į skirtingas grupes, pagal struktūrą ir kompozicijos principą[7]. Vienas iš kompozitinių medžiagų pavyzdžių yra laminatai. Laminatuose stipri ir tvirta medžiaga yra apliejama surišančia medžiaga, kuri padeda išlaikyti formą, užtikrinti apsaugą nuo mechaninių pažeidimų ir perduoti apkrovas stipriai medžiagai. Todėl laminatus galima suskirstyti pagal sustiprinimo būdus (1 pav.)[14].



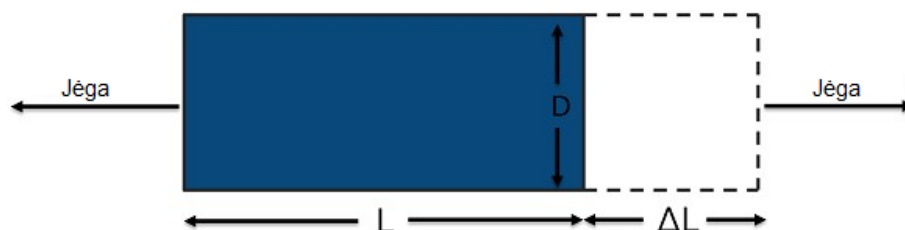
1 pav. Laminatų sustiprinimo būdai: a) ilgi pluoštai b) maži pluoštai c) tekstilės pluoštas d) sluoksnis kitos medžiagos e) išbarstytos dalelės

Rišančioji medžiaga gali būti polimerinė, metalinė ar keramikinė. Tačiau dažniausiai yra naudojama polimerai – dervos arba termoplastikai.

Tačiau kaip ir tradicinės medžiagos, kompozitai patiria jėgas, dėl kurių gali įvykti deformacijos. Kadangi daugelis kompozitinių medžiagų defektų prasideda mikroskopinėje skalėje, todėl yra neįmanoma pamatyti be specialios įrangos[4]. Prasidėjus mikroįtrūkimui, jis pradeda plisti, dėl padidėjusios jėgos likusiai kompozitiniai medžiagai, dėl to kompozitas įtrūkti ar atsisluoksniuoti, taip sukeldamos katastrofinius padarinius. Todėl yra svarbu kompozitinių medžiagų deformacijas, kad būtų galima nuspėti įvykstančią avariją[15].

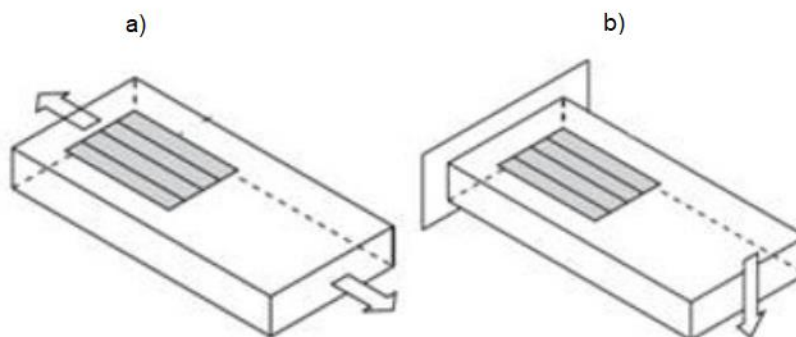
1.2. Kompozitinių medžiagų deformacijų matavimas

Kompozitinių medžiagų deformacijos susiformuoja, kai medžiaga patiria tam tikrą taikomą jėgą[16]. Deformacija yra nusakoma kaip santykis pasikeitusios medžiagos ilgio lyginant su jėgų nepatyrusios medžiagos ilgiu (2 pav.). Deformacija gali būti teigiama, kai medžiaga išsitempia arba neigiama, kai medžiaga susitraukia.



2 pav. Deformacijos ilgio pokytis

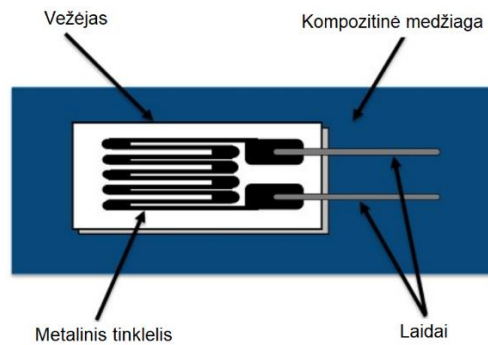
Egzistuoja keturi skirtingi deformacijos tipai: ašinės, lenkimo, šlyties, sukimo[1]. Dažniausiai sutinkamos yra ašinės ir lenkimo deformacijos (3 pav.) Ašinės deformacijos parodo kaip medžiaga plečiasi ar susitraukia, kai ją veikia linijinė jėga, horizontalia ašimi. Lenkimo deformacija įvyksta, kada medžiaga išsiplečia, kai ją veikia linijinė jėga, vertikalia ašimi. Šlyties deformacija įvyksta, kai medžiaga yra veikiamą linijinių jėgų iš abiejų pusių. O sukimo deformacijos įvyksta, kai medžiaga patiria įcentrinės jėgas, tiek vertikalia, tiek horizontalia ašimi.



3 pav. Deformacijos rūšys: a) ašinė b) lenkimo

Deformacijų matavimui dažniausiai yra naudojami įtempių jutikliai, dėl savo mažos savikainos, didelio tikslumo, paprastumo ir lengvumo naudoti ant įvairių paviršių[17]. Įtempių jutikliai keičia elektrinę varžą priklausomai nuo jų veikiančios deformacijos tipo.

Įtempių jutiklis dažniausiai susideda iš labai plono metalo laido arba metalo folijos išdėstytos tam tikra tinklelio struktūra. Tinklelis yra priklijuotas prie plono atraminės plokštelės vadinamos vežėju (angl. *carrier*), kuris tiesiogiai priklijuojamas prie testuojamos medžiagos (4 pav.)



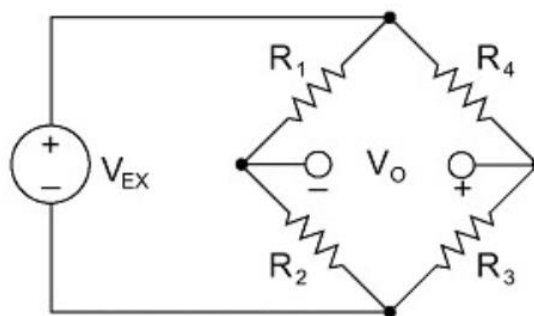
4 pav. Įtempio jutiklio sandara

Pagrindinis parametras įtempių jutiklių parametras yra jo jautrumas deformacijai, kuris išreiškiamas kaip deformacijos faktorius (angl. *gauge factor*). Deformacijos faktorius yra santykis parodantis dalinį pasikeitimą elektrinėje varžoje ir dalinį pasikeitimą medžiagos ilgyje ir yra išreiškimas formule(1):

$$GF = \frac{\Delta R/R}{\Delta L/L} = \frac{\Delta R/R}{\varepsilon} \quad (1)$$

Kur ε – deformacija, ΔL – absoliutus pasikeitimas medžiagos ilgyje, L – originalus ilgis, ΔR – pasikeitimas įtempio jutiklio varžoje, dėl veikiančio deformacijos tipo, R – nedeformuoto įtempio jutiklio varža.

Norint matuoti mažus varžos pasikeitimus įtempių jutikliuose yra naudojamas Vitstono tiltelis (5 pav.)[18] Jis susideda dviejų lygiagrečiai sujungtų rezistorių šakų, kuriuose yra po du nuosekliai sujungtus rezistorius. Šakos sujungtas papildomu laidu, kuriame matuojama įtampa.



5 pav. Vitstono tiltelis skirtas matuoti mažus pasikeitimus varžoje.

Kai paleidžiama V_{EX} įtampa per tiltelį, ir V_0 yra lygus 0, tai formulė (2) parodo santykį tarp dviejų rezistorių šakų.

$$\frac{R_1}{R_2} = \frac{R_3}{R_4} \quad (2)$$

Tai reiškia, jog tiltelis yra subalansuotas. Todėl pasikeitus varžai bent viename rezistoriuje, tiltelis išves ne nuliui lygią reiškią. Todėl, bet kurią varžą tinklelyje pakeitus su įtempių jutikliu, galima pamatyti jo pasikeitimus varžoje ir taip nustatyti deformaciją.

Egzistuoja trys pagrindiniai įtempių jutiklių jungimo konfigūracijos[19]:

- Ketvirtadalis tiltelio – naudojamas tik vienas įtempių jutiklis, kiti pakeičiami rezistoriais atitinkančiais įtempių jutiklio varžą. Dėl to matuoja tik ašinę arba lenkimo deformacijas.
- Pusė tiltelio – naudojami du įtempių jutikliai. Pagal jų išdėstymą galima matuoti ašinę arba lenkimo deformacijas.
- Pilnas tiltelis – naudojami visi keturi įtempių jutikliai. Pagal jų išdėstymą gali matuoti ašinę arba lenkimo deformacijas.

Taip pat svarbi yra nominali įtempių jutiklių varža nedeformuotoje padėtyje. Dažniausiai yra naudojamos varžos yra 120 Ω , 350 Ω ir 600 Ω [20]. Didesnės nominalios varžą, kad sumažinti karštį sugeneruotą tarp lygiagrečiai sujungtų šakų įtampos. Taip pat didesnės nominalinės varžos įtempių jutiklis padeda sumažinti signalo variacijas dėl pasikeitusios temperatūros metaliniame tinkelyje.

Įtempių jutiklis taip pat keičia savo varžumą ir jautrumą nuo pasikeitusios temperatūros, todėl atsiranda paklaida [21]. Kai renkamosi reikėtų rinktis įtempių jutikliai, kurie yra atsparesni temperatūros pokyčiams. Taip pat pasirinkti tiltelio konfigūracija, kuri padėtų kompensuoti temperatūros sukeltus svyravimus.

1.3. Duomenų surinkimo sistemos

Norint nuskaityti duomenys iš įtempių jutiklių reikalinga duomenų surinkimo sistema. Duomenų surinkimo sistema (angl. *DAQ*) yra matavimo sistema ir kompiuteris, gebantis nuskaityti elektrines ar fizines ypatybes ir išsaugoti jas tolimesnei analizei[22]. Tokia sistema susideda (6 pav.) iš trijų pagrindinių dalių: sensoriaus, duomenų surinkimo sistemos įrenginio ir kompiuterio[23].



6 pav. Tipinė duomenų surinkimo sistemos sandara

Sensorius sąveikauja su matuojamu objektu tiesiogiai arba netiesiogiai. Sensoriai pakeičia fizinės reikšmės į elektrinius signalus. Šiame darbe sensoriai yra naudojami įtempių jutikliai, kurie matuoja deformacijas medžiagoje.

Duomenų surinkimo įrenginys yra sistemos subjektas, kuris yra sujungtas tarp sensoriaus ir kompiuterio. Dažnai jis turi keletą įėjimų, todėl vienu metu gali skaityti kelis sensorius. Duomenų surinkimo įrenginys atlieka dvi pagrindines funkcijas: signalo surinkimas ir paverčia analogines

reikšmes į skaitmenines.

Signalų surinkimo metu signalas ne visada yra iškart panaudojamas. Jame gali būti triukšmų, arba signalas būti per silpnas. Todėl prieš perduodant duomenys į kompiuterį, signalas yra išfiltruojamas ir jeigu reikia sustiprinamas.

Analoginis skaitmeninis keitiklis (angl. *ADC*) konvertuoja iš sensoriaus gautas analogines reikšmes į skaitmenines reikšmes, kurios yra perduodamos į kompiuterį.

Paskutinis įrenginys duomenų surinkimo sistemoje yra kompiuteris. Jo tikslas surinkti duomenys iš duomenų surinkimo įrenginio, kad jie būtų išsaugoti tolimesnei analizei. Jame yra įdiegta tvarkyklė kuri leidžia gauti suskaitomus duomenys. Taip pat įdiegta programinė įranga, kurios vartotojo sąsaja lengvai peržiūrėti išsaugotus duomenys ir juos manipuliuoti.

1.4. Įtempių jutiklių duomenų surinkimo sistemų analizė

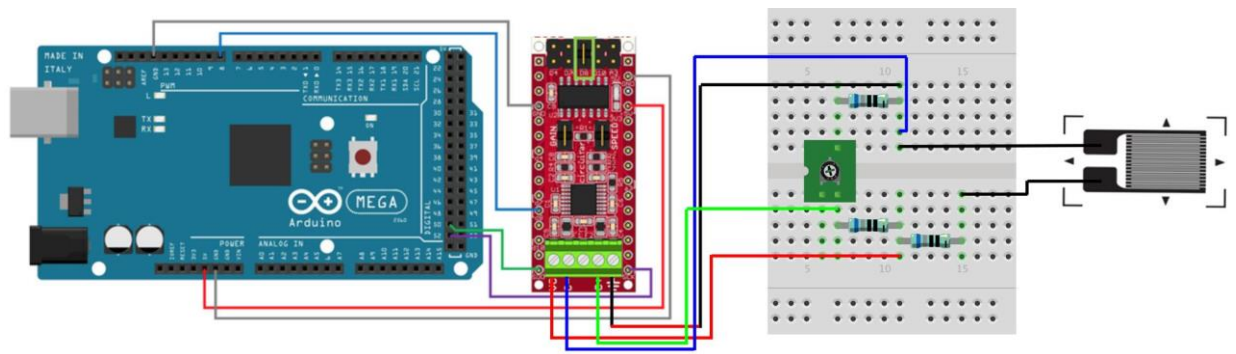
Egzistuoja daugybė duomenų surinkimo sistemų, kurios geba matuoti įvairių įtempio jutiklių gautas reikšmes. Analizės tikslas išanalizuoti duomenų surinkimo sistemas, kurios galėtų tenkinti probleminės srities sprendimą ir atrasti jų privalumus ir trūkumus.

1.4.1. Deformacijų matavimas su ir įtempio jutikliais, naudojantis Arduino mikrovaldikliu

2019 metais, Arthur Guilherme Mereles, Clivaldo Oliveira, José Manoel Balthazar atliko tyrimą, skirtą deformacijų matavimui su įtempių jutikliais, naudojantis Arduino mikrovaldikliu. Jų teigimu įtempio jutikliai yra brangūs ir negalimi panaudoti antrą kartą, todėl eksperimento išlaidos dažnai būna didelės[24].

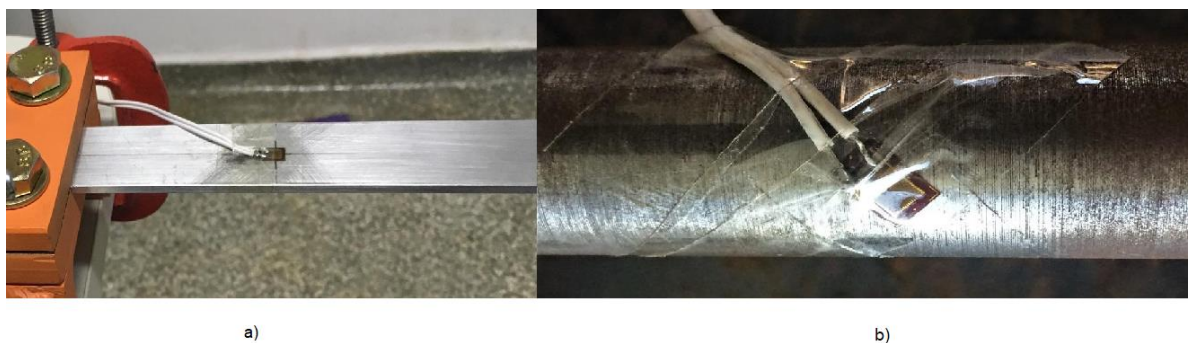
Straipsnyje yra siūloma naudoti Arduino mikrovaldiklį, drauge su mažai kainuojančiais įtempio jutikliais. Tai suteikia privalumų, tokių kaip lengvas implementavimas ir maža pačios sistemos vertė. Straipsnio tikslas sukurti metodologiją, skirtą deformacijų matavimui, naudojantis mažos vertės ir lengvo naudojimosi įrenginius, kurie gali būti naudojamas kaip edukacinis įrankis.

Duomenų surinkimui buvo padaryta sistema (7 pav.) susidedanti iš Arduino Mega2560 R3 mikrovaldiklio ir svarstyklių modulio, kuris paremtas ADS1230 lustu. Jame integruotas stiprintuvas, filtras ir 20 bitų analoginį-skaitmeninį keitiklis, kuris leidžia moduliui konvertuoti gautas analogines reikšmes iš svorio ar įtempių jutiklių į aukštos raiškos skaitmenines reikšmes.



7 pav. Sistemos jungimo diagrama, skirta duomenų surinkimui.

Vėliau buvo atliktas eksperimentas, kurio metu buvo paimti du įtempių jutikliai ir priklijuoti prie skirtingų tiriamų medžiagų. Pirmoje sistemoje, prie sijos įtempių jutiklis buvo priklijuotas pasuktas 90 laipsnių kampu ant horizontaliosios ašies. Antroje sistemoje įtempių matuoklis buvo priklijuotas prie veleno pasuktas 45 laipsnių kampu. (8 pav.) Kadangi sistemose buvo naudojamas Vitstono tiltelis ir tik vienas įtempčių jutiklis, todėl pasirinkta ketvirtadalio tiltelio konfigūracija, kurioje trys varžos elementai buvo pakeisti į fiksuotas reikšmės rezistorius.



8 pav. Įtempių jutiklių priklijavimas prie: a) pirmos sistemos b) antros sistemos

Jo metu buvo matuojamos deformacijos pasiūlytuose sistemose, atliekant po penkis matavimus kiekvienoje iš jų. Buvo gauti 256 išmatuoti rezultatai, kadangi įvesties sparta buvo 80 Hz, tai buvo gauti deformacijos duomenys trunkantis 3 sekundes. Gauti rezultatai pavaizduoti 1 lentelėje:

1 lentelė. Gauti rezultatai matuojant deformacijas

	1 eksperimentinė sistema	2 eksperimentinė sistema
Analitiniai duomenys	208.83	14.02
Apskaičiuoti duomenys	207.98	14.49
Eksperimentiniai duomenys	207.3	13.3

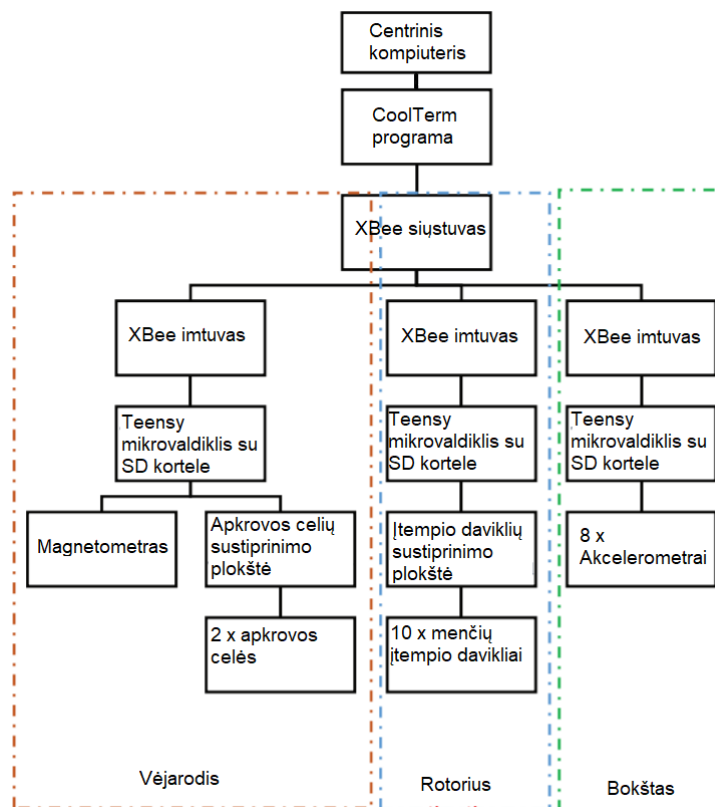
Gauti rezultatai rodo, kad siūloma Gauti rezultatai rodo, kad siūloma sistema pasižymi dideliu tikslumu norimoje pritaikyti srityje, nes gauti rezultatai yra artimi analitiniams ir apskaičiuotiems duomenims.

1.4.2. Mažos kainos, didelio greičio, daugiakanalė Arduino paremta duomenų surinkimo sistema vėjo turbinų sistemoms.

2019 išleistas straipsnis David Bradney, Samuel Evans, Mu Chu and Philip Clausen išleistame straipsnyje siūloma duomenų surinkimo sistemos, paremtos Arduino, nes komerciškai prieinamos sistemos yra brangios ir yra sudėtingai pritaikomos specifinei situacijai[25] Jie sukūrė skirtingas aukšto greičio duomenų surinkimo sistemas, kurių pagalba buvo nuskaityti jutiklių duomenys iš įvairių vėjo jėgainių vietų, tokių kaip rotorius, platforma ir bokštas. Dėl esančių stiprių magnetinių laukų vėjo turbino aikštelėje, turbino pokrypio kryptis ir greitis buvo matuojamas naudojantis GoPro kamera nufilmuotu vaizdu ir analizuojamas Matlab kode. Apdoroti duomenys iš GoPro sistemos buvo suderinti laiku su Arduino sistemos duomenimis, taip leidžiant analizuoti vėjo turbiną.

Straipsnyje buvo sukonstruotos trys atskiros duomenų surinkimo sistemos, skirtos rinkti duomenys iš skirtingų vėjo turbino vietų. Pirmoji ir pati sudėtingiausia sistema, renka įtėpių jutiklių duomenys nuo vėjo turbino menčių, 500 Hz greičiu. Antrosios sistemos tikslas rinkti, 5Hz greičiu, turbino pokrypio krypties ir greičio signalus ir lygiagrečiai apkrovos elementų matuojančių vėjarodžio apkrovą signalus. Trečioji sistema – rinkti duomenys iš akcelerometrų pritvirtintų prie aštuonkampio vėjo jėgainės bokšto. Visų sistemų surinkti duomenys su laiko štampu yra perduodami į centrinį kompiuterį. Duomenys tarp centrinio kompiuterio ir duomenų surinkimo sistemų buvo persiunčiami naudojantis, ZigBee duomenų perdavimo protokolą.

Kiekviena iš trijų duomenų surinkimo sistemų turi unikalių ypatybių (9 pav.)



9 pav. Duomenų surinkimo sistemos blokinė diagrama

Rotoriaus duomenų surinkimo sistema susideda iš Teensy mikrovaldiklio. Jis turi 21 analoginę įvestį, leidžianti prijungti iki 20 įtampių jutiklių. Kadangi buvo skaitoma 500 Hz greičiu, analoginio-skaitmeninio keitiklio raiška buvo nustatyta iki 10 bitų. Tai leido nuskaityti duomenys su pakankama raiška. Mikrovaldiklis buvo duomenys įrašinėjo tiesiogiai į SD atminties kortelę. Įtampių jutikliai buvo prijungti prie Vitstono tiltelio modulių. Juose buvo naudojama ketvirtadalio tilto konfigūracija, tačiau modulyje esantis potenciometras leido tiltelį subalansuoti. Moduliai turi integruotą stiprintuvą, kuris leido signalą sustriprinti 1000. Sumontuoti XBee moduliai, leido tiesioginę duomenų perdavimą į centrinį kompiuterį.

Turbinos pokrypio krypties ir greičio sistema matavimui buvo panaudotas Teensy mikrovaldiklis ir inercijos matavimo modulis. Jis savyje turi integruotą trijų ašių giroskopą, trijų ašių akcelerometrą, ir 3 ašių magnetometrą. Tačiau pokrypio matavimui užtenka magnetometro ir akcelerometro.

Turbinos pokrypio matavimui įtakos turi ir vėjarodis. Todėl ant jo buvo sumontuotos dvi apkrovos celės, kurios buvo matuojamos naudojantis apkrovos celių sustiprinimo plokštė. Gauti duomenys yra išsaugomi į atminties kortelę, ir po to išsiunčiami per XBee modulį į pagrindinį kompiuterį.

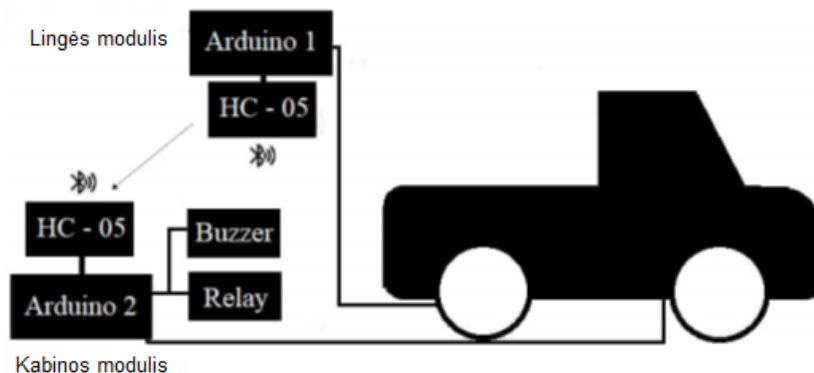
Bokšto sistema susideda iš 8 akcelerometrų sudėliotų aplink aštuonkampio jėgainės bokšto sienas. Tai leido pamatuoti trumpalaikės vibracijos, sukeltas vėjo jėgainės komponento sulūžimo atveju. Sistema taip pat susideda iš Teensy mikrovaldiklio kartu su SD atminties kortelės modeliu. Surinkti duomenys taip pat išsiunčia per XBee modulį į centrinį kompiuterį.

Gauti rezultatai parodė, kad sistema geba rinkti duomenys apie vėjo jėgainės analizei reikalingus duomenys. Pasiūlytas variantas gali būti lengvai išplėstas ir pritaikytas kitoje srityje.

1.4.3. Bluetooth technologija paremta belaidė įtampių matavimo sistema, krovinio transporto perkrovimui stebėti

2019 Masyhur, A. H., Pulung Nurprasetio, I., ir t.t. pasiūlė sprendimo būdą krovinio transporto perkrovimui spręsti. Jų siūloma sistema susideda iš mikrovaldiklio, įtampčių jutiklio ir Bluetooth perdavimo modulio (10 pav.). Ši sistema yra montuoja į patį sunkvežimį ir sistemai pastebėjus perkrovimą yra blokuojamas jos užvedimas [26].

Kad išmatuoti vilkiko svorį, įtampčių jutiklis yra tvirtinimas ant lingės. Kad nuskaityti jutiklio gaunamą informaciją yra naudojamas Arduino mikrovaldiklis, kartu su analoginiu ir skaitmeniniu keitiklio modulio HX711. Jame gaunama 24 bitų skaitmeninė reikšmė. Modulio matavimo greitis buvo nustatytas 10 Hz. Kad nuskaityti įtampčių jutiklio varžą yra naudojamas Vitstono tiltelis ketvirtadalio konfigūracija. Kad išjungti užvedimą Arduino mikrovaldiklis susijungia su kitu mikrovaldikliu per Bluetooth ryšį. Pasirinktas buvo, todėl kad užtikrina maksimalų perdavimą iki 100 metrų ir gali būti užtikrintas ryšys tarp dviejų modulių, ir ryšys negali būti lengvai pertraukiamas.



10 pav. Siūlomos sistemos sandara

Eksperimento metu sistema buvo įdiegta į keturratę transporto priemonę. Atlikus testavimą su svoriu buvo nustatyta 4.4% paklaida, kas leido užtikrinti sistemoje gan tiksliai nustatyti perkrovimą.

1.5. Išanalizuotų duomenų surinkimo sistemų palyginimas ir siūlomi tobulinimo sprendimai

Atsižvelgus į analizes metu išnagrinėtus mokslinius straipsnius apie įtempių jutiklių duomenų surinkimo sistemas, buvo padarytos tokios išvados:

- Pirmame straipsnyje 2019 metais, Arthur Guilherme Mereles, Clivaldo Oliveira, José Manoel Balthazar atliko tyrimą, skirtą deformacijų matavimui su įtempių jutikliais, naudojantis Arduino mikrovaldikliu. Buvo gauta, kad sistema pasižymi dideliu tinsklumu, lyginant su teoriškai gautomis reikšmėmis. Tačiau jų pasiūlytoje sistemoje, susiduria su gan mažu duomenų nuskaitymo greičiu (80 Hz), tačiau aukšta 20 bitų skaitmenine reikšme Sistemoje taip pat susiduriama su limituotu įtempių jutiklių kiekiu. Taip pat yra reikalinga pastovi komunikacija su kompiuteriu.
- Antrame straipsnyje buvo pasiūlytos trys duomenų surinkimo sistemos skirtinguose vėjo jėgainių vietose. Kadangi sistemos buvo sujungtos belaidžiais ZigBee protokolu veikiančiais moduliais, tai leidžia plėsti matuojamą sistemą. Pati sudėtingiausia buvo rotoriaus matavimo sistema. Joje dešimt įtempių jutiklių buvo priklijuoti prie jėgainės mentės. Buvo gautas didelis nuskaitymo greitis (500Hz), tačiau gauta dar mažesnė 10 bitų skaitmeninė reikšmė. Antroji sistema matavo turbinos pokrypio kryptį ir greitį, kartu su vėjarodžio patiriamomis jėgomis. Tam tikslui buvo naudojamas giroskopas, akcelemetras, magnetometras ir apkrovos celės. Duomenys buvo surenkami 5Hz greičiu. Trečioji sistema matavo aštuonkampės jėgainės bokšto sienas patiriamas vibracijas. Jų sukurta sistema parodė, kad Arduino paremtos sistemos duomenų surinkimo sistemos gali veikti ir belaidžiu būdu, net ir sudėtingomis sąlygomis. Tai pat pasiūlyta sistema yra lengvai išplečiama.

- Trečiame straipsnyje buvo pasiūlyta krovinio transporto perkrovimo stebėjimo sistema. Joje įtempčių jutiklis matuoja lingėje įvykstančias deformacijas, taip įvertindamas krovinio svorį. Viršijus nustatytą limitą, sunkvežimyje yra blokuojamas užvedimas. Ši sistema, įrodė, kad įtempčių jutikliai gali būti pritaikomi ir judančiame transporte.

Atsižvelgiant į analizės metu padarytas išvadas, kuriant patobulintą duomenų surinkimo sistemą siūlomi šie sprendimai ir rekomendacijos:

- Vienas įtempčių matavimo mazgas turi turėti galimybę prijungti ne vieną įtempčių jutiklį. Taip galima prijungti Vitstono pusės tiltelio konfigūracija, dėl kurios gaunami tikslesni matavimai.
- Norint pastebėti miesto autobuso kompozitinio kėbulo deformacijas, reikia duomenys iš įtempčių jutikliu nuskaityti ne mažesniu nei 500Hz dažniu ir didesne nei 10 bitų analoginio skaitmeninio keitiklio raiška, kad pastebėti, net ir mažiausias kėbulo deformacijas.
- Sistemoje įtempčių jutiklių duomenys turi būti perduodami bevieliu būdu, norint matuoti miesto autobuso kompozitinio kėbulo deformacijas. Taip pat toks sujungimo būdas sumažina sistemos kompleksškumą ir leidžia lengvą sistemos išplėtimą.
- Įtempčių jutiklių matavimo duomenys turi būti pasiekiami realiu laiku. Tačiau duomenys turėtų būti išsaugomi į vidinę atmintį, kad dingus interneto jungčiai, duomenys išliktų.
- Informacinė sistema turi būti pasiekama ne tik iš kompiuterio, bet ir mobiliųjų įrenginių, naudojantis internetu. Taip sistema gali būti pasiekama iš viso pasaulio, bet kurio paros metu.

2. MIESTO AUTOBUSO KOMPOZITINIO KĖBULO DEFORMACIJŲ SISTEMOS PROJEKTAVIMAS

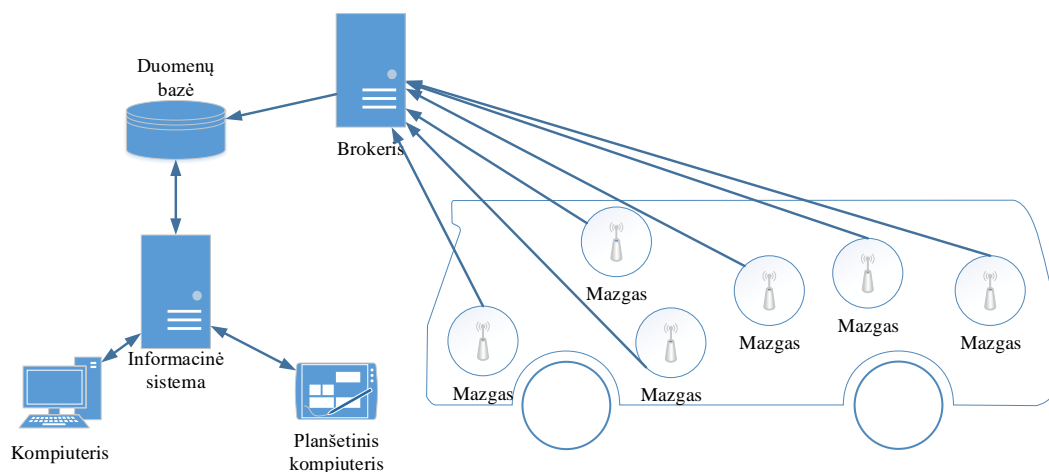
Atlikus analizę buvo išnagrinėtos kompozitinės medžiagos, kodėl svarbu matuoti jų deformacijas ir koku metodu tai daryti. Kaip paaiškėjo analizėje, daugiausia yra naudojami elektrinės varžos įtempių jutikliai. Atlikus mokslinių straipsnių analizę paaiškėjo, kad kompozitinių medžiagų deformacijų matavimo su įtempių jutikliais yra įmanomas, naudojantis lengvai prieinamas elektronikos komponentus. Tokios matavimo sistemos netgi užtikrina tikslius rezultatus, lyginant su suskaičiuotais duomenimis.

Projektavimo tikslas – nubraižyti planuojamos daryti miesto autobuso kompozitinio kėbulo deformacijos sistemos struktūrinę diagramą, išsikelti sistemai taikomus funkcinius ir nefunkcinius reikalavimus, ir nustatyti sistemos vartotojus ir jų roles.

2.1. Miesto autobuso kompozitinio kėbulo deformacijų sistemos modelis

Atsižvelgus į teorinėje dalyje išnagrinėtas duomenų surinkimo sistemas ir pasiūlytus tobulinimo sprendimus, buvo nuspręsta padaryti miesto autobuso kompozitinio kėbulo deformacijų informaciją sistemą, kuri susideda iš belaidžių deformacijos duomenų surinkimo mazgų ir nuotolinės informacinės sistemos.





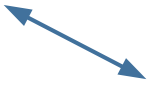

Numatomas sistemos struktūrinė diagrama pavaizduota 11 pav.



11 pav. Kuriamos sistemos struktūrinė diagrama

Struktūrinės diagramos panaudotų simbolių paaiškinimas pateikiamas 2 lentelėje

2 lentelė. Struktūrinės diagramos simbolių paaiškinimas

Simbolis	Paiškinimas
 Mazgas	Kuriamos informacinės sistemos prototipo įtempčių jutiklių deformacijos fiksavimo įrenginys
	Išorė esantis serveris, sukonfigūruotas vykdyti skirtingas sistemoje esančias funkcijas
	Duomenų bazė, joje saugoma įtempčių jutiklių deformacijos ir informacinės sistemos duomenys
	Galutinio vartotojo įrenginiai su kuriais prisijungiama prie informacinės sistemos.
	Rodomas duomenų ryšys abejomis kryptimis.
	Rodomas duomenų ryšys iš-į rodyklės viena kryptimi

Sistemoje mazgas yra atsakingas už deformacijų matavimą miesto autobuso kompozitiniame kėbule. Jis susideda iš mikrovaldiklio, analoginio-skaitmeninio keitiklio, įtempčių jutiklio, Wi-Fi modulio ir išorinės duomenų saugyklos – SD kortelės modulio. Sistema gali būti sudaryta iš didelio skaičiaus mazgų, išdėstytų skirtinguose kėbulo vietose. Duomenys iš mazgo perduodami MQTT protokolu, kuris užtikrina lengvą tinklo išplėtimą, t. y. leidžia pridėti daugiau mazgų. Taip pat nedideliu sunaudojamų duomenų srautu, perduodant duomenis, taip sumažinant energijos suvartojimą.

MQTT yra žinučių eilės sudarymo telemetrijos transporto protokolas (angl. *Message Queuing Telemetry Transport Protocol*). Tai nesudėtingas, atviro kodo protokolas, kuris leidžia galimybę išplatinti informaciją, net ir ribotų išteklių tinkle[27]. Todėl yra tinkamas naudoti tokiais atvejais, kaip komunikacijos mašina su mašina (M2M) ir daiktų internete, kur reikalingos mažos energijos sąnaudos, nedideli skaičiavimo resursai ir tinklo pralaidumas. MQTT yra pagrįstas kliento ir serverio užsiprenumeravimo – leidimo komunikacijos metodu. MQTT serveris vadinamas tarpininku (angl. *broker*), o klientai – tai tiesiogiai prijungti įrenginiai prie jo [28]. Tarpininkas/Brokeris surenka informaciją iš klientų, naudodamasis užsiprenumeruota tema. Taip pat jis kontroliuoja informacijos platinimą. Šio atveju MQTT brokeris, naudojamas duomenų surinkimui iš mazgų ir išsaugojimui

išorėje esančioje duomenų bazėje.

Informacinėje sistemoje gauti duomenys iš duomenų bazės yra atvaizduojami tolimesnei analizei. Išsaugoti duomenys turi būti atvaizduojami tiek grafiškai, tiek skaitine reikšme. Sistemoje turi būti integruota prisijungimo funkcija, tai apsaugo sistemą nuo pašalinių asmenų ir suteikia saugumo.

2.1.1. Funkciniai ir nefunkciniai reikalavimai

Atsižvelgus į ankstesniame skyriuje siūlomą sistemos modelį ir pirmajame skyriuje atliktos duomenų surinkimo sistemų palyginimo analizės išvadų. Buvo suformuoti ir pateikti funkciniai ir nefunkciniai reikalavimai:

- Funkciniai reikalavimai miesto autobuso kompozitinio kėbulo deformacijų informacinei sistemai:
 - Viename mazge turi būti galimybė prijungti daugiau kaip vieną įtempio jutiklį, naudojantis Vitstono tilteliu (pusės arba pilno tiltelio konfigūracijos).
 - Mazgas turi skaityti duomenys iš įtempio jutiklio nemažesniu nei 500 Hz dažniu ir didesne nei 10 bitų analoginio-skaitmeninio keitiklio raiška.
 - Nuskaityti deformacijos duomenys turi būti saugomi į išorinę atminties kortelę, kaip atsarginė kopija.
 - Mazgas turi perduoti duomenys belaidžiu būdu, naudojantis MQTT protokolą.
 - Mazgas turi būti prijungiamas į tinklą savaimiškai.
 - Deformacijų duomenys turi būti saugomi išorėje esančioje duomenų bazėje.
 - Informacinė sistema turi būti prieinama per internetą
 - Informacinė sistema turi turėti prisijungimo funkcionalumą
 - Galimybė turėti vartotojus su skirtingomis teisėmis, kurio galima pridėti, ištrinti, redaguoti.
 - Filtruoti duomenys pagal pasirinktą datą.
- Nefunkciniai reikalavimai miesto autobuso kompozitinio kėbulo deformacijų informacinei sistemai:
 - Mazgo kėbulas turėtų turėti korpusą, kuris turėtų būti atsparus dulkėms, smūgiams ir drėgmei.
 - Sistema turi veikti kompiuteriuose, išmaniuose telefonuose ir planšetiniuose kompiuteriuose, kurie turi prieigą prie interneto.
 - Prie sistemos gali prisijungti neribotas kiekis žmonių.
 - Informacinė sistema turi turėti prisitaikantį dizainą.

2.1.2. Sistemos naudotojai

Kadangi informacinė sistema yra talpinama internete ir lengvai prieinama, todėl buvo padaryta, kad tik sistema gali naudotis tik autorizuoti sistemos vartotojai – administratoriai ir inžinieriai. Skirtumas tarp jų, toks jog administratorius turi prieigą ne tik prie gautų duomenų, bet ir yra atsakingas už sistemoje esančių vartotojų ir deformacijos matavimo mazgų administravimą. Tuo tarpu inžinierius – turi prieigą tik prie sistemoje, esančių išmatuotų duomenų.

3 lentelėje aprašyti administratoriaus atliekamos užduotys ir keliami reikalavimai:

3 lentelė. Sistemos administratoriaus rolė

Naudotojo kategorija	Administratorius
Naudotojo sprendžiami uždaviniai (atliekamos funkcijos):	Stebi deformacijų duomenys, administruoja naujus sistemos vartotojus ir matavimo mazgus.
Patirtis dalykinėje srityje:	Patyręs
Patirtis informacinėse technologijose:	Geros žinios IT srityje
Papildomos naudotojo charakteristikos:	–
Apsimokymo poreikis:	Reikia
Amžiaus grupė:	20–64

4 lentelėje apžvelgiama sistemoje esanti inžinieriaus rolė.:

4 lentelė. Sistemos inžinieriaus rolė

Naudotojo kategorija	Administratorius
Naudotojo sprendžiami uždaviniai (atliekamos funkcijos):	Stebi deformacijų duomenys ir jos analizuoja.
Patirtis dalykinėje srityje:	Patyręs
Patirtis informacinėse technologijose:	Vidutinės IT srityje
Papildomos naudotojo charakteristikos:	–
Apsimokymo poreikis:	Nereikia
Amžiaus grupė:	20–64

Į vartotojų roles, bus atsižvelgiama kuriant informacinę sistemą.

2.2. Miesto autobuso kompozitinio kėbulo deformacijų sistemos prototipas

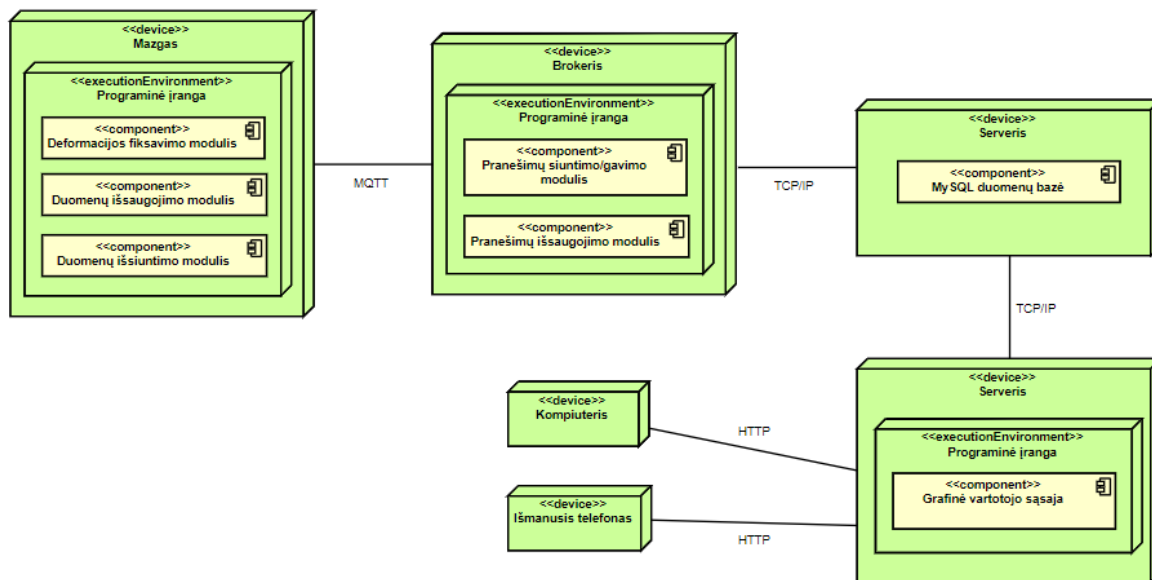
Pagal anksčiau atliktos analizę, kurios metu pasiūlyti deformacijų matavimų sistemų tobulinimo būdai ir išsikeltus funkcinius ir nefunkcinius reikalavimus. Buvo pradėtas konstruoti miesto autobuso kompozitinio kėbulo deformacijos sistemos prototipas. Konstruojamą sistemą galima išskirstyti į dvi dalis: deformacijų fiksavimų ir deformacijų atvaizdavimo sistemos prototipus.

2.2.1. Deformacijų fiksavimų prototipas

Atsižvelgus į anksčiau nubraižytą struktūrinę schemą, deformacijų fiksavimų prototipas susideda iš daugybės išdėstytų mazgų, brokerio ir duomenų bazės. Mazgo tikslas rinkti deformacijų

fiksavimų duomenys, naudojantis įtempių jutikliu ir gautus duomenys išsaugoti vidinėje atmintyje ir tuo pačiu metu persiųsti tinklu į brokerį. Brokeris gavęs duomenys juos išsaugo į duomenų bazę. Duomenų bazėje saugoma išsaugomos deformacijos reikšmės, tolimesnei jų analizei.

Buvo nubraižyta kuriamo deformacijų fiksavimo prototipo diegimo diagrama, naudojant UML notaciją. Joje atspindės techninės įrangos komponentai ir naudojami programiniai komponentai [29] (12 pav.)



12 pav. Miesto autobuso kompozitinio kėbulo deformacijų fiksavimų sistemos prototipo diegimo diagrama

Pagal diegimo diagramą ir pasiūlytus tobulinimo sprendimus mazgo prototipo kūrimui buvo pasirinkti šie komponentai:

ESP8266

Šis modulis, gaminamas „Ai-Thinker“ gamintojo. Tai vienas iš daugelio šios firmos gaminamų modulių naudojančių ESP8266 lustą.

ESP8266 yra WiFi lustas su pilnu TCP/IP protokolų palaikymu ir su integruotu mikrovaldikliu, gaminamas Kinijos gamintojo „Espressif Systems“. Pagrindinės ESP8266 lusto charakteristikos [30]:

- 32-bitų Tensilica Xtensa L106 procesorius, veikiantis 80 MHz greičiu.
- 64 kB instrukcijų RAM
- 96 kB duomenų RAM
- 64 kB paleisties ROM
- Išorinė „Flash“ atmintis nuo 512 kB iki 4 MB.
- IEEE 802.11 b/g/n Wi-Fi palaikymas
- 16 įvesties/išvesties lizdų
- SPI, I2C, I2S, UART protokolų palaikymai
- 10 bitų analoginis-skaitmeninis keitiklis.

„Ai-Thinker“ gamintojo gaminami moduliai, naudojantys ESP8266 lustą yra vadinami ESP-XX. Šiuo metu paskutiniosios laidos yra ESP-201 modulis. 5 lentelėje pateikiami pagrindiniai skirtumai, tarp skirtingų modulių

5 lentelė. ESP modulių skirtumai

Pavadinimas	Naudojami lizdai	Antena	Matmenys (mm)	Ekranuotas?	„Flash“ atmintis
ESP-01	6	Integruota PCB	14.7 × 12.1	Ne	512kB
ESP-02	6	Išorinė jungtis	14.2 × 14.2	Ne	512kB
ESP-03	10	Keramikinė	14.2 × 14.7	Ne	512kB
ESP-04	10	Nėra	20.0 × 16.0	Ne	512kB
ESP-05	3	Išorinė jungtis	17.0 × 16.0	Ne	512kB
ESP-06	11	Nėra	10.0 × 10.0	Taip	512kB
ESP-07	14	Keramikinė ir išorinė jungtis	14.2 × 10.0	Taip	1MB
ESP-08	10	Nėra	17.3 × 12.1	Taip	-
ESP-09	10	Nėra	24.0 × 16.0	Ne	1MB
ESP-10	3	Nėra	24.0 × 16.0	Ne	512kB
ESP-11	6	Keramikinė	24.0 × 16.0	Ne	512kB
ESP-12	14	Integruota PCB	24.0 × 16.0	Taip	4MB
ESP-12E	20	Integruota PCB	18.0 × 20.0	Taip	4MB
ESP-12F	20	Integruota PCB	24.3 × 16.2	Taip	4MB
ESP-12S	20	Integruota PCB	14.7 × 12.1	Taip	4MB
ESP-13	16	Integruota PCB	14.2 × 14.2	Taip	4MB
ESP-14	22	Integruota PCB	14.2 × 14.7	Taip	-
ESP-201	22	Integruota PCB	33.5x25.5	Ne	512kB

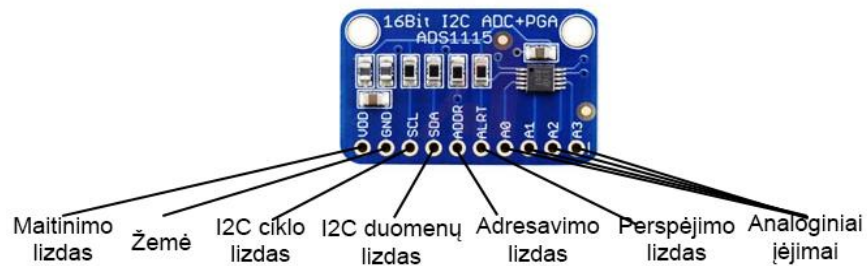
Kadangi prototipe ESP8266 naudojamas ne kaip WiFi modulis prie išorinio valdiklio, o kaip mikrovaldiklis, todėl buvo svarbios pasirinktos charakteristikos. Buvo pasirinktas ESP-12E modulis, nes turi reikiamą kiekį lizdų (20), didelę atmintį (4 MB) ir palaiko SPI ir I2C protokolus, kurie bus reikalingi komunikacijai su kitais projekte naudojamais moduliais.

ADS1115

Analoginis skaitmeninis keitiklis prototipe yra skirtas įtempių jutiklių duomenų nuskaitymui. Buvo pasirinktas ADS1115 analoginis skaitmeninis modulis (13 pav.) nes jis turi tokias galimybes ir funkcijas[31]:

- Mažas modulio dydis (2mm x 1.5mm x 0.4mm)
- Platus įtampos diapazonas, nuo 2.0 V iki 5.0V.
- Mažos energijos sąnaudos nuolatinio režimu apie 150 uA.
- Programuojamas duomenų perskaitymo dažnis nuo 8Hz iki 860Hz.
- Geba užtikrinti 16 bitų analoginio-skaitmeninio keitiklio raišką.
- Turi I2C sąsaja.

- Turi programuojamą stiprintuvą.
- 4 analoginius arba 2 diferencinius analoginius įėjimus.

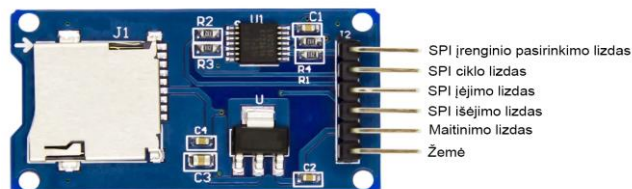


13 pav. Analoginio skaitmeninio keitiklio modulio išdėstymas

Prototipo kūrimui buvo pasirinktas šis modulis, nes veikia su mikrovaldikliu suderinama 3.3 voltų įtampa, taip pat palaikomu I2C komunikacijos protokolu. Nors ESP8266 mikrovaldiklis turi integruotą analoginį skaitmeninį keitiklį, tačiau jis tėra 10 bitų, todėl pagal anksčiau pasiūlytus sistemos tobulinimo sprendimus, pasirinktas modulis geba skaityti 16 bitų raiška ir didesniu negu 500 Hz duomenų perskaitymo greičiu. Taip pat modulis turi galimybę prijungti įvairių konfigūracijų Vitstono tiltelių.

SD kortelės modulis

Duomenų išsaugojimui buvo pasirinktas SD kortelės modulis (14 pav.), nes mikrovaldiklis neturi pakankamai atminties, kad išsaugoti didelį kiekį nuskaitytų duomenų. Todėl buvo pasirinktas šis modulis, kuris veikia su 3.3 voltų įtampa, komunikuoja per SPI protokolą ir geba palaikyti iki kortelės iki 16 GB vietos[32].



14 pav. SD kortelės modulio išdėstymas

Nors duomenys prototipe yra išsiunčiami MQTT protokolu į brokerį, kuris išsaugo duomenys duomenų bazėje, kortelėje yra saugomi deformacijų matavimai, kaip atsarginė kopija nepavykus persiųsti duomenų.

Įtempių jutiklis

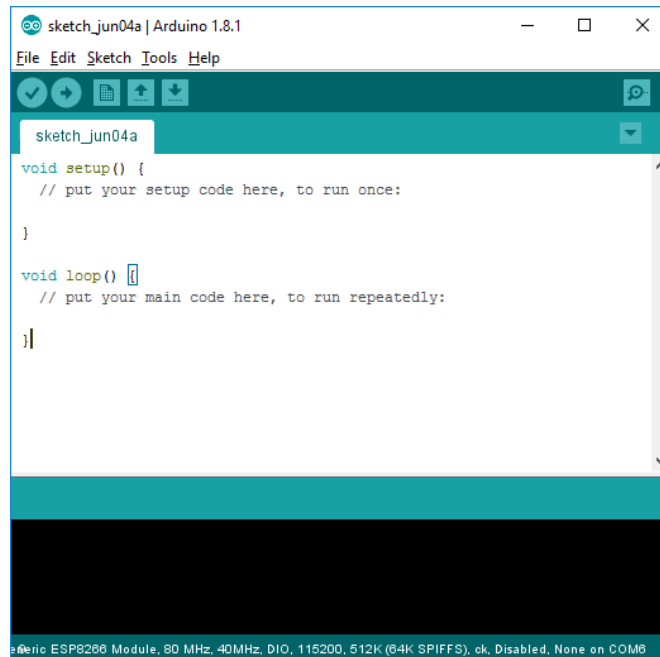
Deformacijų fiksavimo prototipui buvo pasirinktas rezistorinis įtempių jutiklis BX120-20AA (15 pav.). Jo išorės išmatavimai yra 20 x 3 mm ir nominali įtempių jutiklio varža 120 Ω ir deformacijos faktorius – 2. Jutiklio bazė pagaminta iš fenolio dervos, o metalinis tinklelis pagamintas iš vario ir nikelio lydinio [33].



15 pav. Naudojamo įtempių jutiklio struktūra

Prototipe įtampos jutiklis sujungiamas ketvirtadalio Vitstono konfigūracija. Tai reiškia, kad tiltelyje yra vienas įtempių jutiklis ir trys 120 Ω atitinkamos varžos. Gauti tiltelio deformacijų fiksavimai yra siunčiami į analoginį skaitmeninį keitiklio modulį, tolimesniam duomenų apdorojimui.

Įkelti kodui į mikrovaldiklį ir į ESP12-E modulį buvo pasirinkta „Arduino IDE“ programinė įranga (16 pav.). „Arduino IDE“ yra kompiuterinė programa, kuri leidžia parašyti programinį kodą, jį kompiliuoti ir įkelti į mikrovaldiklį[34]



16 pav. Arduino IDE pagrindinis programos langas.

Įkėlimo metu C++ kalba parašytas kodas yra perduodamas „avr-gcc“ kompiliatoriui, kuris paverčia kodą, suprantamą mikrovaldikliui. Šio funkcionalumo metu yra paslepiama daug niuansų susijusių su mikrovaldiklių programavimu, todėl yra dažnai naudojamas prototipų gamyboje, dėl lengvo panaudojimo. Naujausia Arduino IDE versija yra 1.8.12, kuri veikia tiek su Windows, Linux ir MacOS operacinėmis sistemomis.

Arduino IDE aplinkoje, sukurti projektai vadinami: „sketch“. Jos sudaro dvi pagrindinės funkcijos (15 pav.):

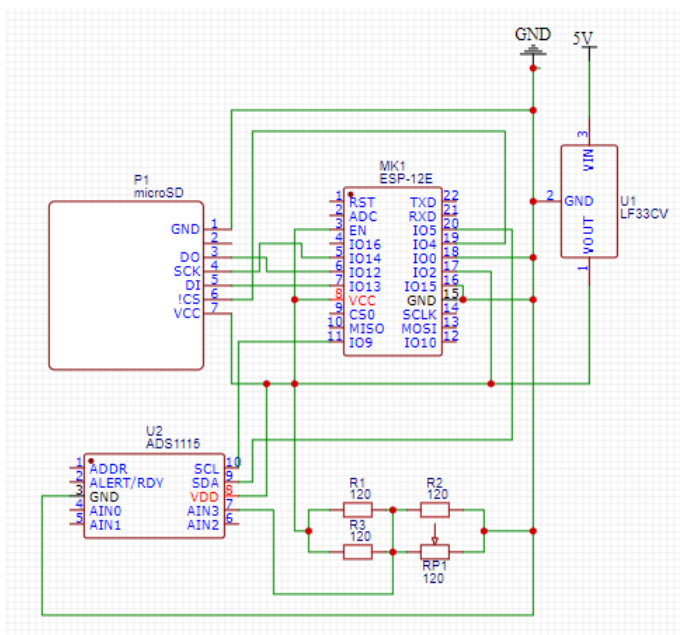
- Funkcija „Setup“.
- Funkcija „Loop“.

Esminis skirtumas tarp funkcijų yra tai, kad funkcija „Setup“ yra vykdoma vieną kartą, kai mikrovaldiklis yra įjungiamas arba perkraunamas. Šioje funkcijoje yra inicijuojami kintamieji, bibliotekos, lizdų nustatymai. Funkcija „Loop“ yra vykdoma po „Setup“ funkcijos ir yra nuolat pasikartojanti. Tai leidžia mikrovaldikliui reaguoti į pasikeitusius išorės veiksnius per atitinkamus sensorius.

MQTT brokeriui sukurti buvo naudojama „Mosquitto“ programinė įranga. Tai atviro kodo programinė įranga suteikianti standartus atitinkantį serverį ir klientų implementacijas[35]. MQTT naudoja publikavimo/prenumeravimo modelį, sunaudoja mažai tinklo resursų ir gali būti įdiegtas į mažai energijos suvartojančius IoT įrenginius. „Mosquitto“ susideda iš trijų dalių:

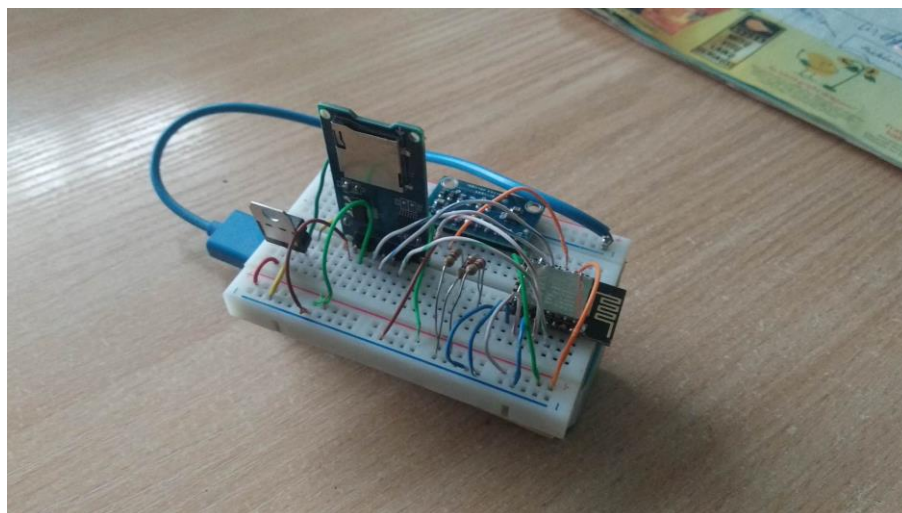
- Pagrindinio „Mosquitto“ serverio
- „Mosquitto“ publikavimo ir prenumeravimo klientai.
- MQTT biblioteka parašyta C programavimo kaina.

Mazgo prototipo kūrimui buvo panaudoti anksčiau paminėti komponentai ir programinė įranga. Naudojantis nemokamu įrankiu internete „EasyEDA“ buvo sumodeliuota mazgo elektroninė schema (17 pav.)



17 pav. Deformacijų fiksavimo mazgo prototipo elektroninė schema

Remiantis mazgo elektrine schema, buvo sukonstruotas vieno mazgo prototipas, naudojantis bandomąja plokšte (18 pav.)



18 pav. Surinktas deformacijų fiksavimo mazgo prototipas

Surinkus mazgo prototipą buvo parašytas programinis kodas. Kodas pateikiamas 1 priede, ir jis buvo parašytas ir įkeltas, naudojantis „Arduino IDE“ programine įranga. Programos kodą galima išskaidyti į dvi pagrindines dalis:

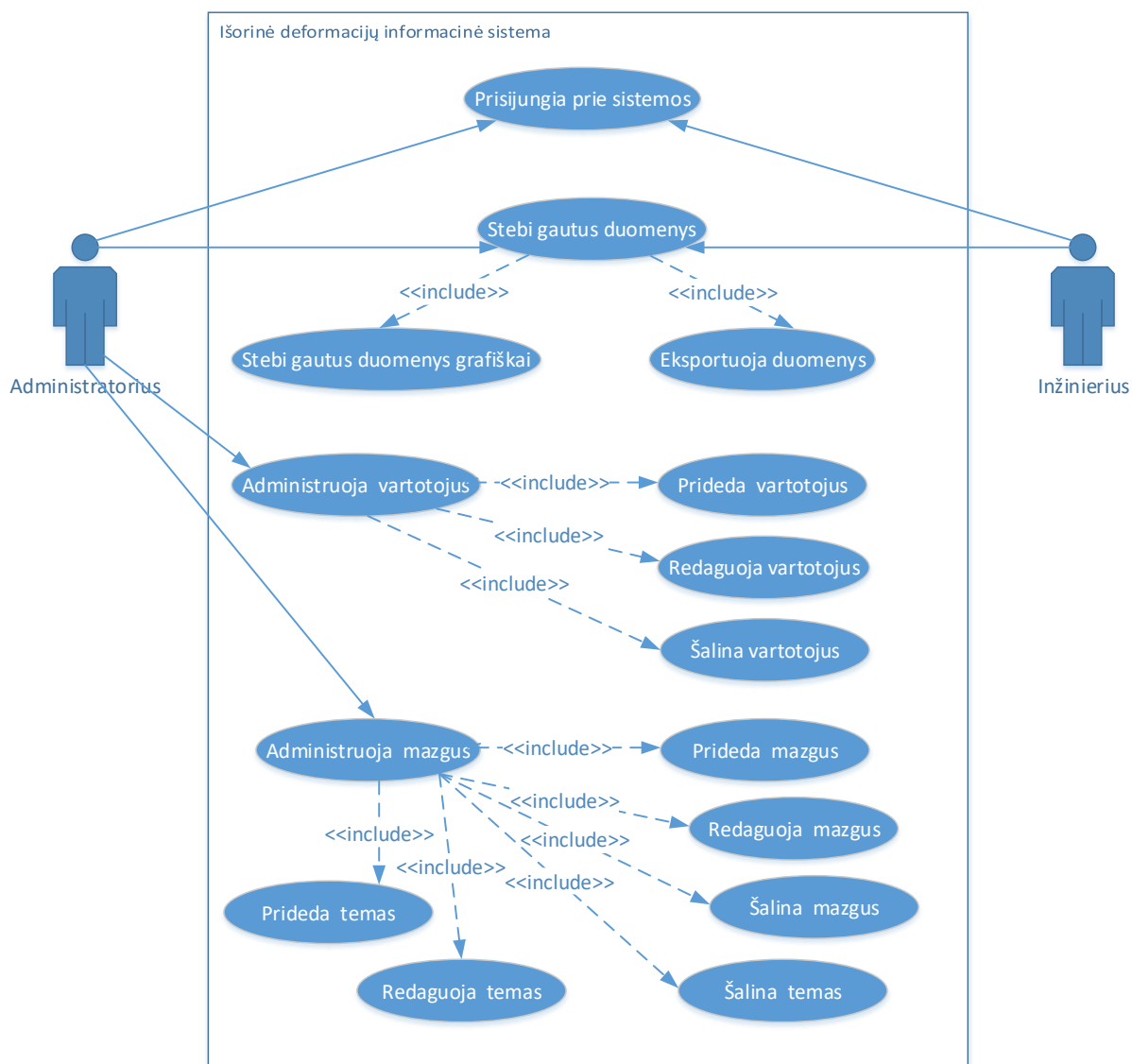
- Deformacijos fiksavimo
- Duomenų išsaugojimo ir perdavimo MQTT protokolu.

Deformacijos fiksavimo metu, naudojantis Adafruit_ADS1X15 biblioteka (https://github.com/adafruit/Adafruit_ADS1X15), yra nuskaitoma įtempių jutiklio reikšmė ir ji padidinama du kartus su integruotu stiprintuvu. Gauta reikšmė yra išsaugoma į SD kortelėje esantį tekstinį failą, ir po to išsiunčiama MQTT kliento biblioteka (<https://github.com/knolleary/pubsubclient>) į brokerį, su tema – gauge/x/xxx. Pirmas žodžiu apibrėžiama, kad tai yra įtempių jutiklio duomenys, raidė x - pakeičiama atitinkamu mazgo numeriu, kad duomenys būtų galima atskirti iš kokios kėbulo vietos jie buvo surinkti, o raidės xxx – siunčiamos temos pavadinimas. Temos leidžia siųsti kelių rūšių duomenys iš vieno mazgo.

Suprogramavus mazgą, kompiuteryje buvo paleistas MQTT brokeris, naudojantis „Mosquitto“ programine įranga ir Python kalba parašytas kodas, kuris išsaugo gautą žinutę į duomenų bazę. Python kodas pateikiamas 2 priede. Jame gautos MQTT žinutės nuskaitymui yra naudojama Paho-mqtt biblioteka (<https://pypi.org/project/paho-mqtt/>). Jos pagalba, galima gauti žinutės iš MQTT brokerio ir MySQL Python Connector bibliotekos (<https://pypi.org/project/mysql-connector-python/>), kuri leidžia tiesiogiai komunikuoti su išorėje esančia duomenų baze.

2.2.2. Deformacijų informacinė sistema

Atsižvelgus į ankščiau išsikeltus sistemos funkcinis ir nefunkcinis reikalavimus ir nustatytus sistemos naudotojus buvo apibrėžtos sistemos panaudojimo ribos. Tam tikslui buvo nubraižyta panaudos atvejų diagrama, kurio pavaizduota aktorių (vartotojų) ir sistemos sąveika[36].



19 pav. Panaudos atveju diagrama

Pagrindinis kuriamos išorinės deformacijų informacinės sistemos tikslas pasiekti iš mazgų surinktas deformacijų matavimus ir juos atvaizduoti lengvai žmogui suprantama forma. Atsižvelgus į anksčiau išnagrinėtą duomenų surinkimo sistemų analizę, sistemai buvo pasiūlyti tokie sprendimo būdai,:

- Sistema turi būti pasiekiamą, bet kurio paros metu ir bet kurioje pasaulio vietoje.
- Sistema turi būti suderinama su išmaniaisiais įrenginiais, tokie kaip mobilusis telefonas, planšetė. Taip pat su kompiuteriais.
- Nereikalinga įdiegti papildoma programinė įranga.
- Deformacijų fiksavimų duomenys turi būti pasiekiami realiu laiku.

Norint įgyvendinti panaudos atveju diagramoje, aprašytas sąveikas ir anksčiau išsikeltus sistemos tobulinimo sprendimus buvo nuspręsta naudoti LAMP – tai yra akronimas susidedantis iš WEB sistemose, naudojamų pavadinimų: Linux, Apache, MySQL, PHP. Tai dažniausiai naudojama atviro kodo programinė įranga internetiniuose serveriuose, dėl didelio aptaranuojamo užklausų

skaičiaus, sugebėjimo tvarkytis su didelėmis reliacinės duomenų bazėmis ir serverio resursų tausojimo[37].

Kuriant WEB sistemą dažnai norima ją padaryti kuo greičiau ir efektyviau [38]. Tačiau tai įgyvendinama, tik turint gerą sistemos struktūrą. Kadangi LAMP pakete yra naudojama PHP programavimo kalba, todėl yra naudojami jai sukurti karkasai. Karkase, programavimo procesas yra standartizuotas ir automatizuotas. Todėl yra automatiškai sugeneruojamos sistemos galimybės tokios kaip apsaugos užtikrinimas, užklausų į duomenų bazę gavimas ir duomenų atvaizdavimas. Tai leidžia programuotojams labiau fokusuotis į verslo logikos implementaciją.

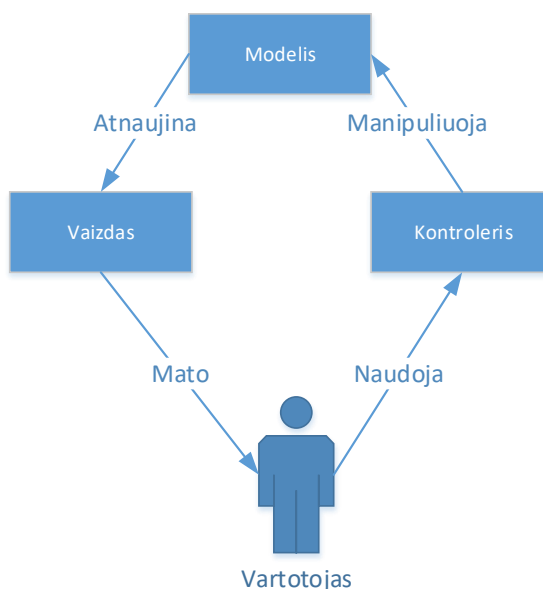
Išorinės deformacijų informacinės sistemos kūrimui buvo pasirinktas Laravel PHP karkasas, dėl savo populiarumo. Remiantis Google Trends paieškos informacija per 1 metus laiko visame pasaulyje, daugiausia užklausų yra apie Laravel karkasą.



20 pav. Google Trends paieškų skaičius apie PHP karkasus

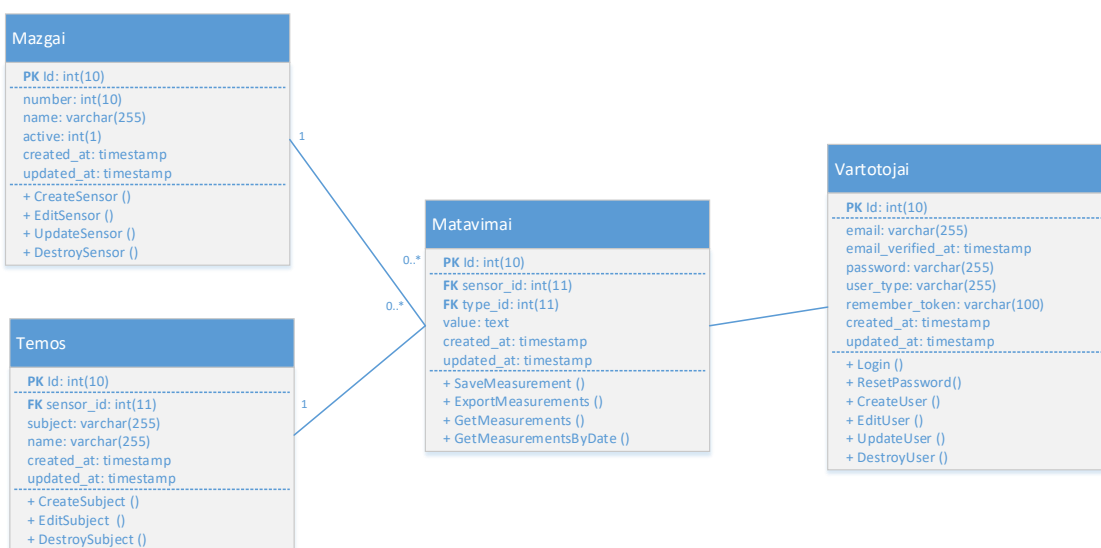
Laravel yra MVC modeliu paremtas, WEB sistemų programavimo karkasas, parašytas PHP kalba. Jis buvo sukurtas, kad pagerinti kuriamos sistemos kodo kokybę, skatindamas naudoti kodą daug kartų, taip sumažinant programavimo laiką [39]. Taip užtikrinimas kodo moduliškumas, nes įvairios funkcijos, tokias kaip autentifikacija, prieiga prie duomenų bazės ar sesijos valdymas jau yra suprogramuotos ir ištestuotos. Taip pat karkase yra integruota tam tikra sistemos aplankalų struktūra, kad atskirti karkaso dalis ir taip sutaupyti dar daugiau laiko programuojant.

MVC modelis karkase naudojamas tam, kad atskirti atvaizdavimo duomenys nuo funkcijų, skirtų jų išgavimui. Šis modelis susideda iš trijų pagrindinių dalių: modelis, vaizdas ir kontroleris (20 pav.)[40]. Jame modelis yra atsakingas su duomenimis susijusią logiką, tai gali būti duomenys, kurie siunčiami vaizdui arba kontroleriui ar kiti su verslo logika susiję duomenys. Vaizde yra naudojamas naudotojo sąsajos logikai. Kontroleris yra naudojamas kaip sąsaja tarp modelio ir vaizdo komponentų. Jame yra apdorojama verslo logika, įeinančios užklausos, manipuluojami duomenys iš modelio ir perduodami vaizdui, galutiniam atvaizdavimui.



21 pav. MVC modelio schema

Kuriamos išorinės informacinės sistemos struktūra buvo atvaizduota klasių diagramoje (22 pav). Joje matoma, kad visos sistemos funkcionalumas, susideda iš 4 klasių. Vartotojų klasė yra atsakinga už tokius vartotojo veiksmus, kaip prisijungimas prie sistemos, ar slaptažodžio atkūrimas. Taip pat ši klasė apima ir vartotojų administravimą: naujų vartotojų kūrimą, jų informacijos redagavimą ar ištrynimą. Mazgo klasė apibrėžia fizinio matavimo mazgo pavadinimą, unikalų numerį ir aktyvumo statusą. Taip pat naujų mazgų pridėjimą, redagavimą ir ištrynimą. Temos leidžia iš vieno mazgo priimti skirtingus duomenys. Todėl joje galima pridėti naujas temas, jas redaguoti, ištrinti ir priskirti mazgą unikalų numerį iš mazgo klasės. Matavimų klasėje yra išsaugomi matavimai, su mazgu ir tam tikra tema. Šioje klasėje galima nuskaityti ir filtruoti gautas deformacijų fiksavimo reikšmes.

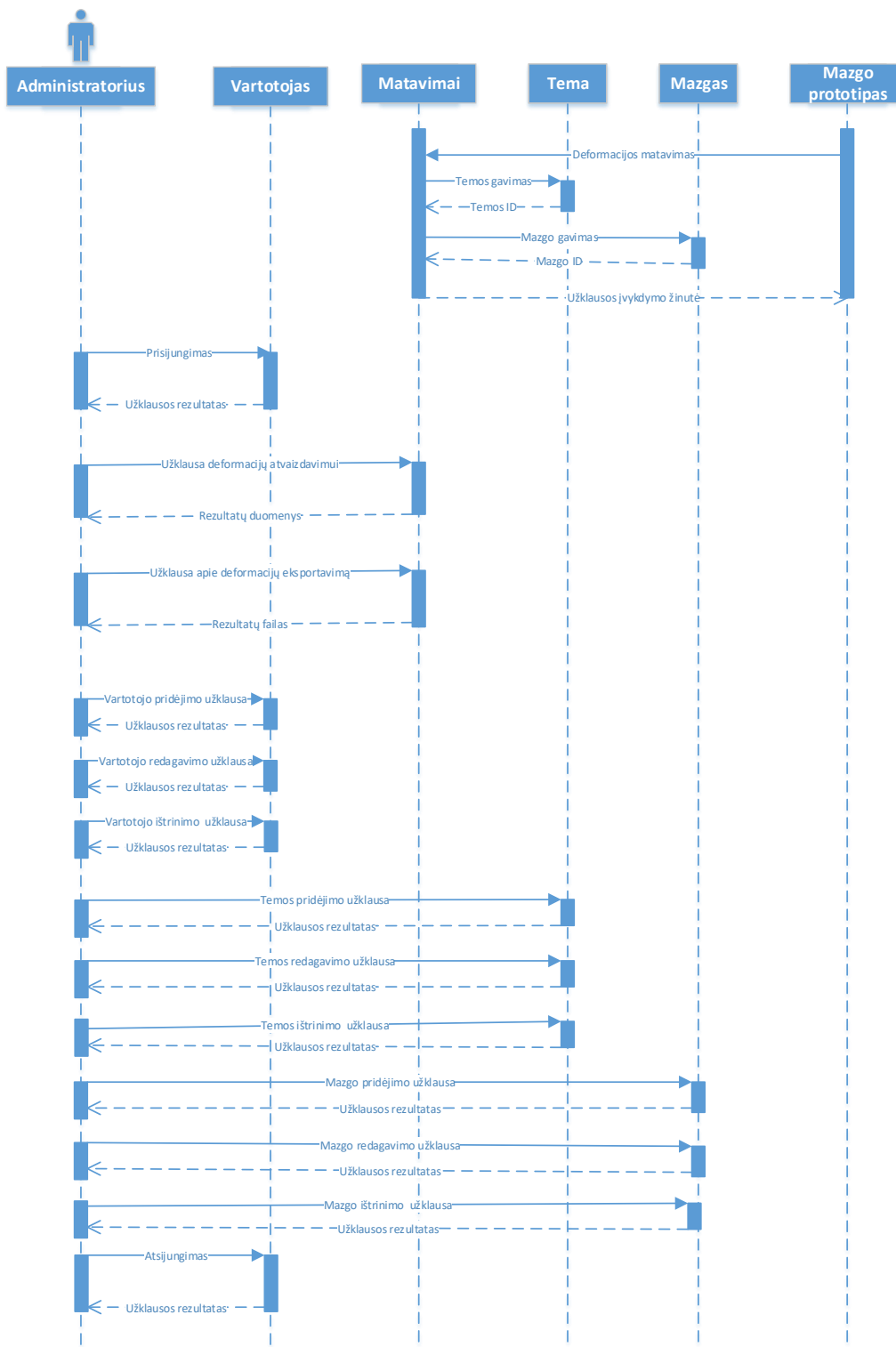


22 pav. Sukurtos išorinės deformacijų sistemos klasių diagrama

Kuriama išorinė deformacijų informacinė sistema su vartotoju sąveikauja per maršruto klasę,

kurioje vartotojui informacija iš duomenų bazės yra pateikiama per HTTP protokolą. Vartotojas siunčia atitinkamą užklausą į maršruto klasę, kuri pasiekia reikiamą klasę ir grąžina rezultatus vartotojui.

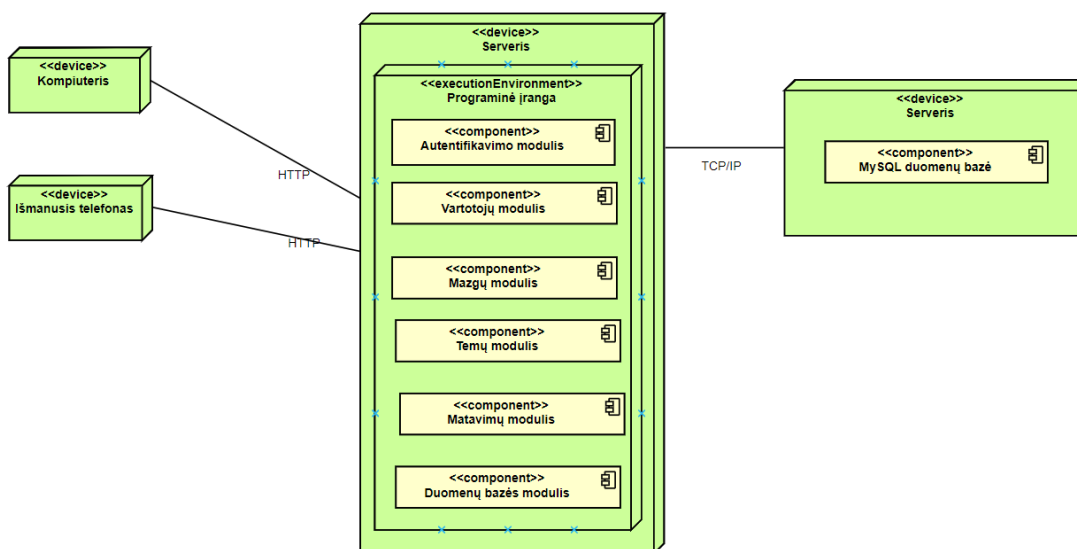
Kuriamos informacinės sistemos sąveikai buvo nubraižyta sekų diagrama (23 pav.). Šioje diagramoje yra pavaizduojama standartinė su sistema atliekamų veiksmų seka su išorėje esančiais sistemos aktoariais[41].



23 pav. Sukurtos išorinės deformacijų sistemos sekų diagrama

Deformacijų matavimo mazgas, išsiunčia duomenys į išorę esančią informacinę sistemą, kuri išsaugo duomenys, duomenų bazėje, taip pat priskirdama unikalų informacinės sistemos mazgo ir temos kodus. Prisijungęs prie sistemos administratorius, gali atlikti tokius pagrindinius veiksmus, kaip deformacijų fiksavimų duomenų peržiūra ir jų eksportavimas, temų administravimas, vartotojų administravimas ir mazgų administravimas. Norint baigti darbą yra atsijungia ir apie tai gaunamas pranešimas ar užklausa įvykdyta sėkmingai.

Taip kaip ir deformacijų fiksavimo prototipo kūrimo buvo nubraižyta išorinės deformacijų informacinės sistemos diegimo diagrama. Joje matyti techninės įrangos komponentai ir naudojami programiniai komponentai (24 pav.)



24 pav. Kuriamos informacinė sistemos diegimo diagrama

Iš diegimo diagramos matyti, jog sistema yra tiesiogiai pasiekama per įvairius įrenginius, tokius kaip kompiuteris, išmanusis telefonas ar planšetė. Sistemos viduje galima išskirti keturis pagrindinius modulius. Vartotojų modulis atsakingas už naujų vartotojų kūrimą, redagavimą ir saugojimą. Mazgų modulyje – priskiriamas unikalus mazgo prototipo numeris, jam suteikiamas atpažinimui skirtas pavadinimas ir jo aktyvumo statusas. Temų modulio pagalba yra galimybė rinkti kelis matavimo parametrus iš to pačio mazgo prototipo. Jame priskiriamas unikalus temos pavadinimas.

O likusieji du moduliai yra Laravel karkase integruoti sprendimai. Jų pagalba be papildomo programavimo buvo integruotas autentifikavimo modulis, kurio pagalba yra prisijungiama prie sistemos ir nustatoma jam priskirta rolė. Palei tai nusprendžiama, kokius pagrindinius modulius jis gali pasiekti. Taip pat naudojamas ir duomenų bazės modulis, kurio pagalba yra išgaunami duomenys iš duomenų bazės, jau suprogramuotomis funkcijomis.

Remiantis anksčiau nubraižytomis diagramos buvo sukurta informacinė deformacijų atvaizdavimo sistemos prototipas. Jis buvo sukurtas, naudojantis Laravel karkasu, kurio versija 5.8.16 ir MySQL duomenų bazės versija – 5.7.2.4. Vartotojo sąsajai kurti buvo pasirinktas Bootstrap 4

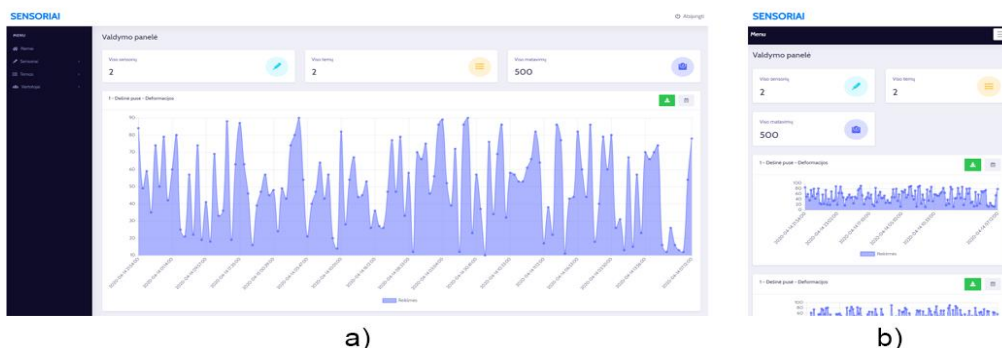
versijos karkasas. Jis susideda iš didelės kolekcijos patogių ir pernaudojamą kodo dalių, parašytų naudojantis HTML, CSS, Javascript programomis. Jo pagalba galima greitai sukurti prisitaikančio dizaino vartotojo sąsajas. Jis apjungia tokias vartotojo sąsajos funkcijas, kaip konteineriai, stulpeliai, lentelės, mygtukai, perspėjimai, formos, sąrašai, navigacijos ir t.t.

Pirmiausia įėjus į sistemą yra patenkama į prisijungimo langą, kuriame galima prisijungti suvedus gautus duomenis arba atkurti pamirštą slaptažodį (25 pav.).



25 pav. Išorinės deformacijų informacinės sistemos pradinis langas

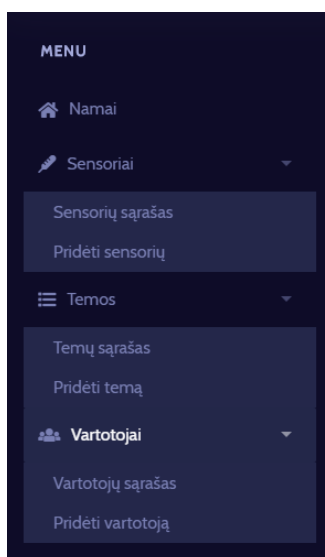
Prisijungus į patenkama į sistemos skydelį (26 pav.), kuriame yra pateikiama informacija apie veikiančių sensorių ir temų kiekį. Taip pat apie pateikiamas bendras deformacijų matavimo skaičių. Žemiau rodoma deformacijų fiksavimo mazgo prototipo gauti duomenys realiu laiku, su galimybe jos filtruoti ir eksportuoti. Kaip ir buvo minėta anksčiau sistema yra prisitaikančio dizaino, todėl veikia ir su išmaniaisiais įrenginiais.



26 pav. Išorinės deformacijų informacinės sistemos skydelis a) kompiuteriuose b) planšetėje

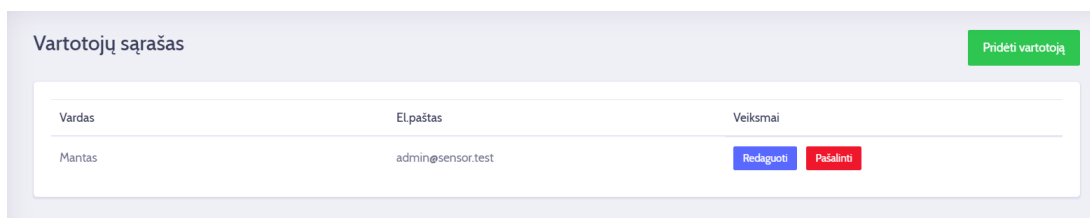
Remiantis išorinės deformacijų informacinės sistemos klasių diagrama, trys klasės (mazgų, temų ir vartotojų) yra atvaizduotos meniu juostoje (27 pav.) į kurias įėjus galima atlikti aprašytas

funkcijas, tokias kaip pridėjimas, redagavimas ar ištrynimas



27 pav. Išorinės deformacijų informacinės sistemos išskleistas meniu sąrašas

Vartotojų klasėje buvo įgyvendintos klasių diagramoje numatytos funkcijos, tokios kaip vartotojo sukūrimas, redagavimas, ištrynimas. Naujo vartotojo pridėjimo ir egzistuojančio vartotojo redagavimo formose yra užpildomi tokie laukeliai, kaip vardas, el.paštas, prisijungimo slaptažodis ir jo rolė. Kaip ir buvo minėta anksčiau, sistemoje yra dvi rolės: vartotojo ir administratoriaus. Pagrindinis skirtumas tarp jų tai atliekamos funkcijos. Vartotojas gali tik peržiūrėti deformacijų fiksavimų duomenis, o administratorius – administruoti vartotojus, mazgus ir jų temas. Vartotojų administravimo langas pateiktas 28 pav.



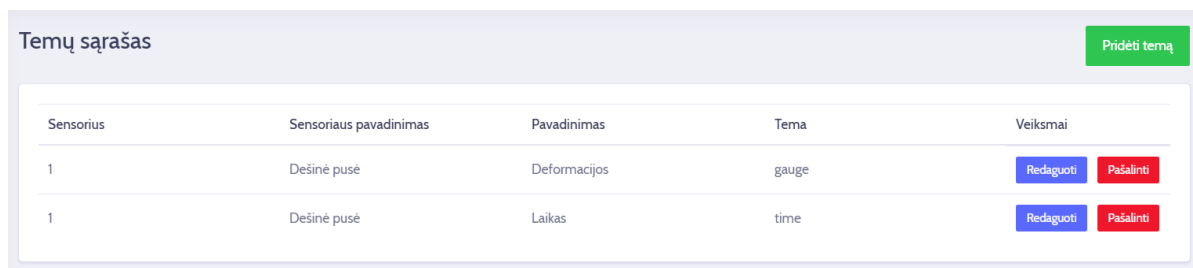
28 pav. Išorinės deformacijų informacinės sistemos vartotojų administravimo langas

Mazgų klasėje taip buvo įgyvendintas jų administravimas. Jis susideda iš mazgų pridėjimo, redagavimo ir pašalinamo. Prie mazgo yra įvedamas unikalus mazgo prototipo numeris, jo pavadinimas, kad užtikrinti vartotojui lengvesnį duomenų supratimą ir jo statusą – aktyvus arba ne. Mazgo pridėjimo/redagavimo forma yra pateikiama 29 paveiksluke.



29 pav. Išorinės deformacijų informacinės sistemos mazgo pridėjimo/redagavimo langas

Sukūrus mazgų administravimą, buvo pereita prie jų temų administravimo funkcionalumo sukūrimo. Temos skirtos, norint surinkti skirtingo tipo informaciją iš vieno to pačio mazgo. Sistemoje taip pat kaip ir mazgų administravime, jas galima sukurti, redaguoti ir ištrinti. Kuriant temą yra pasirenkamas unikalus mazgo prototipo numeris, pagal kurį gaunamas mazgo pavadinimas. Taip pat įvedamas pavadinimas, skirtas vartotojo lengvesniam supratimui ir unikali tema, pagal kurią atskiriama informacija, pradiniam skydelyje. Temų administravimo langas yra pavaizduotas 30 paveiksliuke.



Sensorius	Sensoriaus pavadinimas	Pavadinimas	Tema	Veiksmai
1	Dešinė pusė	Deformacijos	gauge	Redaguoti Pašalinti
1	Dešinė pusė	Laikas	time	Redaguoti Pašalinti

30 pav. Išorinės deformacijų informacinės sistemos mazgo temų administravimo langas

3. MIESTO AUTOBUSO KĖBULO DEFORMACIJŲ SISTEMOS EKSPERIMENTIS TYRIMAS

Sukurtą miesto autobuso kėbulo deformacijų informacinę sistemą sudaro deformacijų fiksavimo mazgas ir informacinė sistema. Deformacijų fiksavimo mazgas, fiksuoja deformacijas įvykstančias kompozitiniame autobuse kėbule. Duomenys yra siunčiami MQTT protokolu į išorę esantį brokerį, kuris duomenys išsaugo į tai pat išorėje esančią duomenų bazę. Informacinė sistema yra pasiekama internetu, bet kurio paros metu, bet kurioje vietoje ir iš skirtingų rūšių įrenginių. Kadangi sistema susideda iš dviejų skirtingų dalių: deformacijų fiksavimo mazgo ir informacinės sistemos, todėl testavimas buvo atliktas individualiai kiekvienai sistemos daliai.

3.1. Deformacijų fiksavimo mazgo testavimas

Deformacijos fiksavimo mazgo prototipo testavimas buvo atliktas remiantis 12 paveikslėliu, kuriame pavaizduota diegimo diagrama. Joje galima įžvelgti, jog mazgas susideda iš trijų programinės įrangos modulių:

- Deformacijos fiksavimo modulis
- Deformacijos išsaugojimo modulis
- Deformacijos išsiuntimo modulis

Todėl kiekvienas programinės įrangos modulis buvo ištestuotas atskirai.

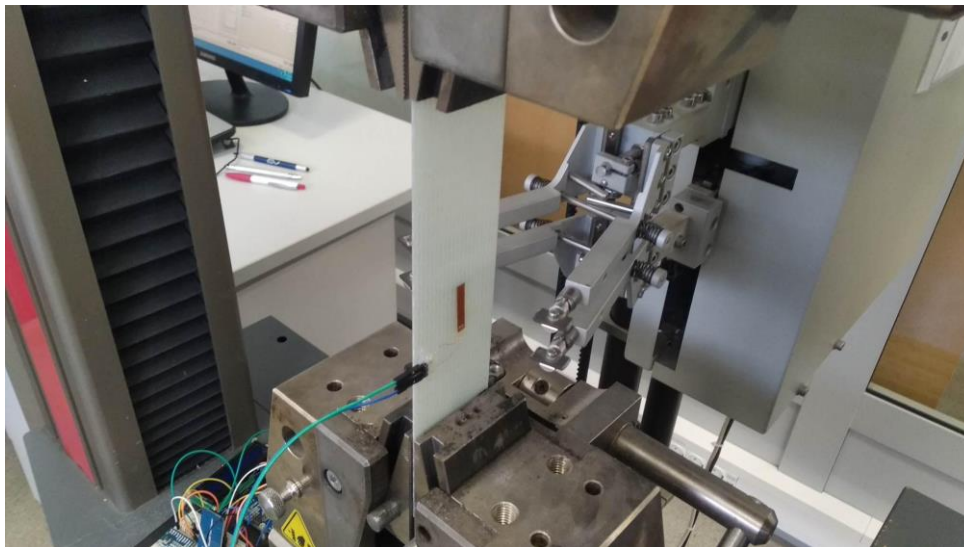
3.1.1. Deformacijų fiksavimo modulio testavimas

Šio testo tikslas – palyginti deformacijos fiksavimo prototipo gautas deformacijos reikšmes su komerciniu deformacijų matuokliu. Šiam testui atlikti prie miesto autobuso kompozitinio kėbulo bandinio buvo priklijuotas vienas įtempių jutiklis (31 pav.), kuris prie sistemos buvo prijungtas, naudojantis ketvirtadalio Vitstono tiltelio konfigūracija.



31 pav. Kompozitinio kėbulo testavimo bandinys

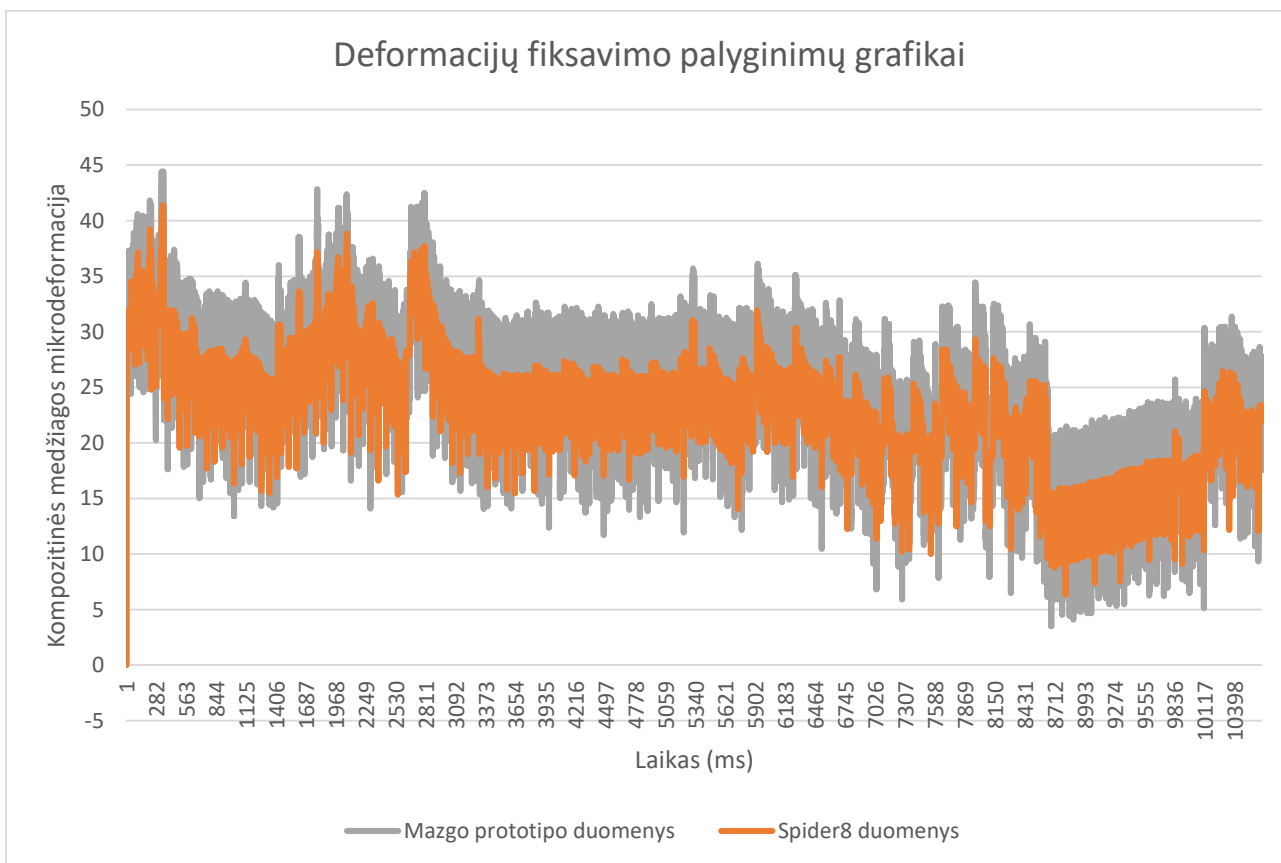
Testavimo bandinys buvo įstatytas į stakles, kuriose jis buvo tempiamas vertikalia ašimi (32 pav.).



32 pav. Kompozitinio kėbulo bandinio testavimas staklėse

Įtempių jutiklių duomenys buvo nuskaityti su deformacijos fiksavimo mazgo prototipu. Po to buvo atliekamas tas pats testas tik duomenys buvo nuskaityti su komerciniu įtempių matuokliu – Spider 8.

Gauti duomenys buvo atvaizduoti ir palyginti grafike (33 pav.)



33 pav. Deformacijų fiksavimo palyginimo grafikai

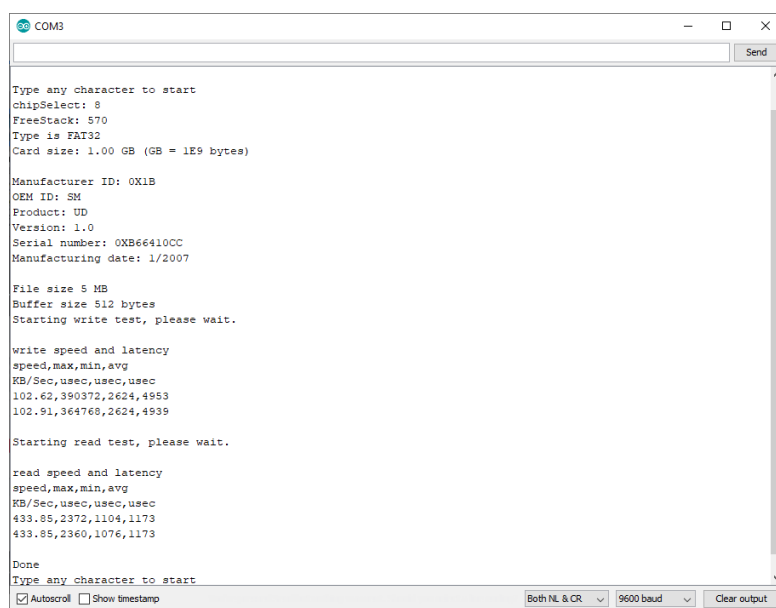
Horizontalioje ašyje yra pateikiamas laikas milisekundėmis, o vertikalioje užfiksuota mikro deformacijos reikšmė, kuri buvo apskaičiuota formule [42]:

$$\varepsilon = 4 \frac{V_o}{V_{EX}} \frac{1}{GF} \quad (3)$$

Iš grafikų buvo apskaičiuota, kad mazgo užfiksuoti duomenų paklaida yra apie 5 procentus lyginant su komerciniu įtempių matuokliu duomenimis. Tai reiškia, kad sukurtas mazgo prototipas gali būti naudojamas deformacijoms fiksuoti.

3.1.2. Deformacijos fiksavimo išsaugojimo modulio testavimas

Deformacijų fiksavimų išsaugojimo modulio testavimo tikslas – ištestuoti įrašymo ir nuskaitymo greitį į SD kortelės modulį. Testui atlikti buvo panaudota „SdFat“ biblioteka (<https://github.com/greiman/SdFat>). SD kortelės modulio testavimas pavaizduotas 34 paveikslėlyje.



```
COM3
Type any character to start
chipSelect: 8
FreeStack: 570
Type is FAT32
Card size: 1.00 GB (GB = 1E9 bytes)

Manufacturer ID: 0X1B
OEM ID: SM
Product: UD
Version: 1.0
Serial number: 0XB66410CC
Manufacturing date: 1/2007

File size 5 MB
Buffer size 512 bytes
Starting write test, please wait.

write speed and latency
speed,max,min,avg
KB/Sec,usec,usec,usec
102.62,390372,2624,4953
102.51,364768,2624,4939

Starting read test, please wait.

read speed and latency
speed,max,min,avg
KB/Sec,usec,usec,usec
433.85,2372,1104,1173
433.85,2360,1076,1173

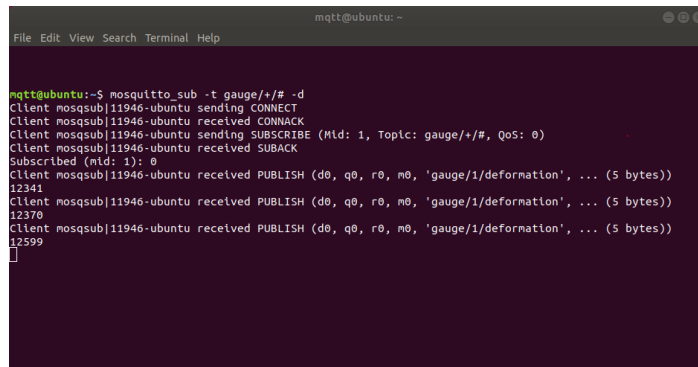
Done
Type any character to start
 Autoscroll  Show timestamp
Both NL & CR 9600 baud Clear output
```

34 pav. SD kortelės modulio testavimas

Buvo gautas 102.62 kb/s įrašymo greitis ir 433.85 kb/s nuskaitymo greitis. Tai reiškia, kad sistemos mazgas yra pajėgi išsaugoti didelius informacijos srautus, ateinančius iš deformacijų fiksavimo modulio.

3.1.3. Deformacijos fiksavimo išsiuntimo modulio testavimas

Deformacijos fiksavimų išsiuntimo modulio testavimo tikslas – ištestuoti duomenų perdavimą iš prototipo į serverį, duomenų išsaugojimui. Kadangi duomenų perdavimas vyksta MQTT protokolu, todėl duomenys priima serveryje esantis „Mosquitto“ programinės įrangos brokeris [43]. Duomenų priėmimo testavimas atvaizduotas 35 paveikslėlyje.

A terminal window titled 'mqtt@ubuntu: ~' showing the output of the 'mosquitto_sub' command. The logs show a client connecting, subscribing to the topic 'gaUGE/+/#', and receiving three PUBLISH messages with 5 bytes of data. The messages are timestamped 12341, 12370, and 12599.

```
mqtt@ubuntu:~$ mosquitto_sub -t gauge/+/# -d
Client mosqsub|11946-ubuntu sending CONNECT
Client mosqsub|11946-ubuntu received CONNACK
Client mosqsub|11946-ubuntu sending SUBSCRIBE (Mid: 1, Topic: gauge/+/#, QoS: 0)
Client mosqsub|11946-ubuntu received SUBACK
Subscribed (mid: 1): 0
Client mosqsub|11946-ubuntu received PUBLISH (d0, q0, r0, m0, 'gauge/1/deformation', ... (5 bytes))
12341
Client mosqsub|11946-ubuntu received PUBLISH (d0, q0, r0, m0, 'gauge/1/deformation', ... (5 bytes))
12370
Client mosqsub|11946-ubuntu received PUBLISH (d0, q0, r0, m0, 'gauge/1/deformation', ... (5 bytes))
12599

```

35 pav. Išsiųsti duomenys į MQTT brokerį.

Iš paveikslėlio matyti, jog brokeris sėkmingai priima duomenys iš mazgų. Vadinasi mazgas išsiunčia informaciją sėkmingai. Taip pat matyti, kad naudojant MQTT protokolą vienas matavimas užima 5 baitų. Tai reiškia, kad norint išsiųsti tokį kiekį duomenų yra sunaudojama mažai energijos [44]

3.2. Informacinės sistemos testavimas

Šio testavimo metu yra patikrinama kiekviena kuriamos sistemos funkcija su jai iškelto reikalavimu ir specifikacija. Šiam testui buvo pasirinktos sistemos funkcijos iš anksčiau sudarytos panaudos atvejų diagramos (19 pav.). Todėl buvo ištestuotos, šios sistemos funkcijos:

- Prisijungimas prie sistemos
- Slaptažodžio atkūrimas
- Naujo vartotojo sukūrimas
- Vartotojo redagavimas
- Vartotojo ištrynimasis
- Naujo mazgo pridėjimas
- Mazgo redagavimas
- Mazgo ištrynimasis
- Temos sukūrimas
- Temos redagavimas
- Temos ištrynimasis
- Deformacijų fiksavimų peržiūra
- Deformacijų fiksavimų išeksportavimas
- Deformacijų fiksavimų filtracija

Atlikus testavimą išsiskelto sistemos funkcijos buvo įgyvendintos sėkmingai. Taip pat sistemoje visos nuorodos veikia taisyklingai. Įvedus neteisingą nuorodą, sistemos vartotojas yra grąžinamas į pradinį lapą. Buvo išbandyta ir formų verifikacija. Jos metu yra tikrinama ar laukeliai

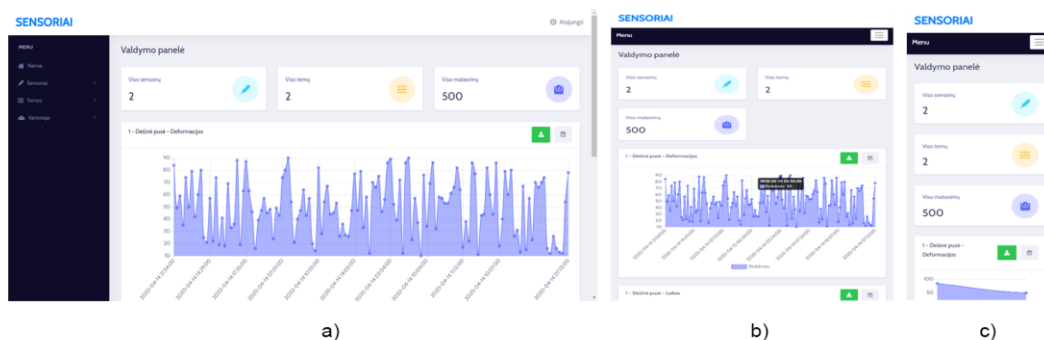
nėra tušti ir jie nepažeidžia jiems iškeltų taisyklių. Įvykus formos verifikacijai vartotojas yra perspėjamas raudoname fone esančiu klaidos pranešimu, dėl ko galėjo įvykti klaida.

Buvo atliktas ir sistemos suderinamumo testas. Jos metu tikrinama suprogramuotos informacinės sistemos suderinamumas su skirtingais įrenginiais ir naršyklėmis [45]. Jo metu patikrinama ar tekstas yra lengvai perskaitomo dydžio, lengvai paspaudžiami mygtukai ir neatsiranda klaidų susijusių su vartotojo patirtimi.

Sistemos suderinamo testavimui buvo pasirinkti trys įrenginiai:

- Nešiojamas kompiuteris su Windows 10 1909 versijos operacine sistema su interneto naršyklėmis:
 - Google Chrome versija 83.0.4103.61
 - Mozilla Firefox 76.0.1
 - Microsoft Edge 44.18362.449.0
- Planšetė Samsung Galaxy Tab A, kurioje įdiegta Android 9.0 versija ir buvo testuojama su interneto naršyklėmis
 - Google Chrome versija 83.0.3359.158
 - Samsung Internet Browser versija 11.1.1.52
- Išmanusis telefonas Xiaomi Redmi 5 su Android 8.1 versija ir įdiegtos interneto naršyklės:
 - Google Chrome versija 81.0.4044.138
 - Xiaomi Browser versija 10.4.8

Testavimo metu buvo bandoma anksčiau išsikeltas sistemos funkcijas. Ištestavus su minėtais įrenginiais ir naršyklėmis, naršymas po sistemą buvo sklandus ir be trūkumų. Tekstai buvo lengvai perskaitomi ir nuorodos nukreipdavo į reikiamas vietas. 36 paveikslėlyje pavaizduotas sistemos panelės langas: a) atvaizduotas nešiojame kompiuteryje, kurio ekrano skiriamoji geba 1920x1080 b) atvaizduotas planšetiniame kompiuteryje, kurio skiriamoji geba – 1280x800, c) išmaniajame telefone, kurio skiriamoji ekrano geba – 720x1440



36 pav. Išorinės deformacijų informacinės sistemos skydelio langas, skirtinguose įrenginiuose

3.3. Miesto kompozitinio kėbulo deformacijų fiksavimo prototipo bandymas

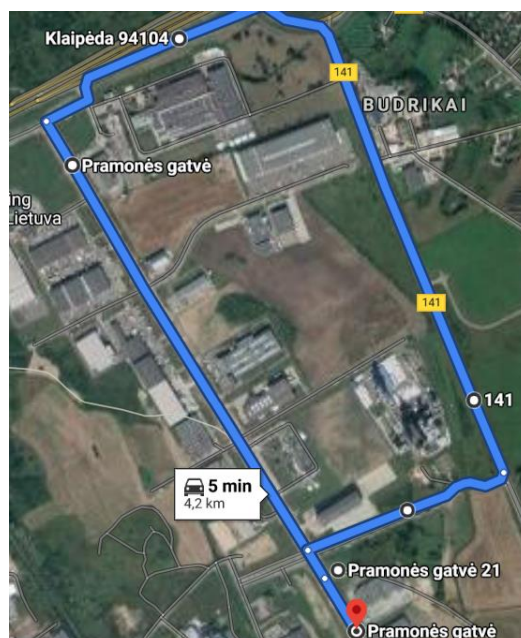
Atlikus pavienius modulių testavimus, buvo atliktas deformacijos duomenų surinkimas elektriniame mieste autobuse (33 pav.).



37 pav. Sistemos prototipo bandymui buvo pasirinktas elektrinis miesto autobusas „Dancerbus“

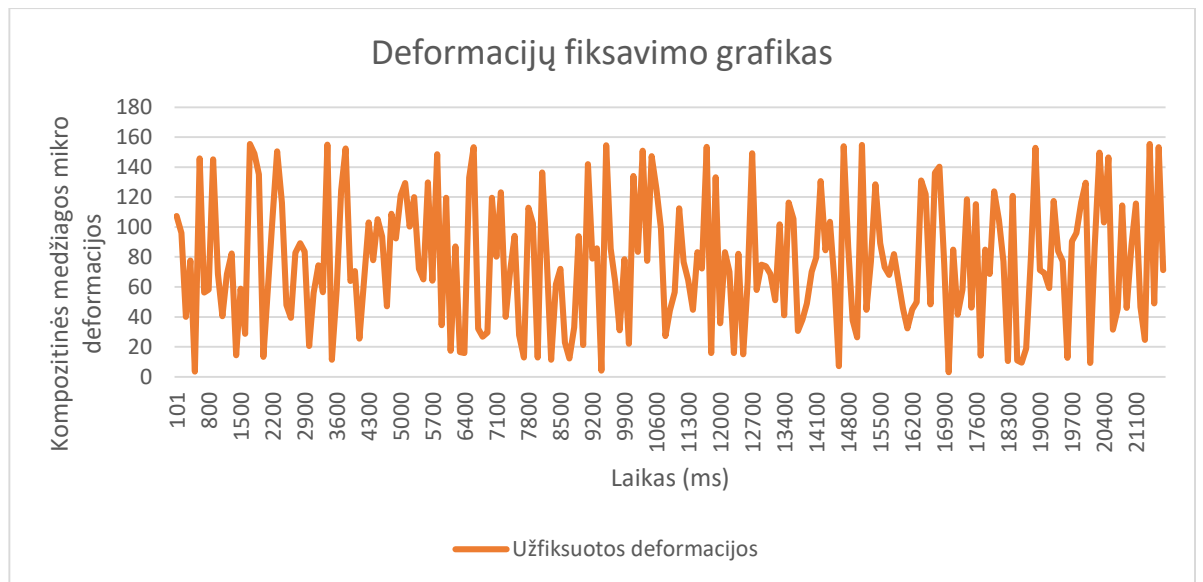
„Dancerbus“ – tai elektrinis miesto autobusas, sukurtas Klaipėdoje. Lyginant su konkurentais, autobusas yra lengvesnis, nes pasižymintis kompozitiniu kėbulu, dėl kurio jis sveria 9775 kg. Autobusas talpina 90 žmonių ir gali nuvažiuoti 90 kilometrų nuotolį. Taip pat jis gali išvystyti iki 70 km/h greitį.

Įtempimų jutiklis buvo prijungtas prie miesto autobuso kompozitinio kėbulo, prie jo buvo prijungtas deformacijų matavimo mazgo prototipas, naudojantis ketvirtadalio Vitstono tiltelio konfigūraciją. Miesto autobusu prototipu buvo padarytas bandomasis važiavimas 4.2 kilometro ilgio.



38 pav. Bandomojo važiavimo maršrutas

Maršruto įveikimas truko 7 minutes ir buvo pririnkta 218820 matavimo duomenų. Tai reiškia, jog matavimo metu pavyko išgauti 521 Hz matavimo greitį. Gauti duomenys buvo atvaizduoti grafike.



39 pav. Testavimo metu surinktų deformacijų fiksavimo duomenys

Grafike deformacijos reikšmės buvo gautos, panaudojus 3 formulę. Iš grafiko matoma, kad nors autobusas patiria mikro deformacijas, tačiau jos nebuvo stiprios, dėl kurių galėtų prasidėti įtrūkimas ir pasižeisti autobuso kėbulas.

IŠVADOS

1. Išanalizavus mokslinius straipsnius susijusius su kompozitinių medžiagų deformacijų fiksavimu buvo išsiaiškinta, jog naudojantis nebrangiu mikrovaldikliu ir rezistoriniais įtempių jutikliais galima tiksliai matuoti medžiagoje vykstančias deformacijas lyginant su apskaičiuotomis reikšmėmis. Taip pat buvo išsiaiškinta, jog panašaus tipo sistemos gali būti sėkmingai integruotos rinkti duomenims sudėtingomis sąlygomis, tai yra vėjo jėgainėse. Analizės metu taip pat buvo išsiaiškinta, jog panašaus tipo sistemą galima pritaikyti ir judančiame transporte, šio atveju tai sunkvežimio blokavimo sistema, persikrovus. Kuriant patobulintą deformacijų fiksavimo sistemą buvo pasiūlyta naudoti įtempių jutiklius, sujungtus Vitstono tilteliu, naudoti didesnę negu 500Hz deformacijų fiksavimo dažnį su didesne nei 10 bitų analoginiu-skaitmeninio keitiklio raiška. Taip pat deformacijas matuojantys mazgai duomenys turi perduoti belaidžiu tinklu, taip leisdami galimybę išplėsti matavimo tinklą.
2. Projektavimo metu buvo nubraižyti planuojamos daryti miesto autobuso kompozitinio kėbulo deformacijos sistemos struktūrinė diagrama, iškelti funkciniai ir nefunkciniai sistemos reikalavimai, nustatyti sistemos vartotojus. Tai buvo atlikta remiantis anksčiau atlikta analize ir joje iškeltus sistemos tobulinimo pasiūlymus
3. Atlikus planuojamos kurti miesto autobuso kompozitinio kėbulo deformacijos sistemos projektavimą buvo pradėtas kurti jos prototipas. Kadangi sistema susideda iš deformacijų fiksavimo ir deformacijų informacinės sistemos. Deformacijų fiksavimo prototipui buvo parinktos, tokios dalys kaip ESP8266 mikrovaldiklis, ADS1115 analoginis skaitmeninis keitiklis, SD kortelės modulis, įtempių jutiklis. Deformacijos informacinės sistemos prototipas buvo sukurtas WEB pagrindu, todėl jos kūrimui buvo panaudotas Laravel PHP kalbos karkasas ir Bootstrap 4 karkasas, apjungiantis HTML, CSS ir Javascript technologijas
4. Sukurtiems prototipams buvo atliktas testavimas. Deformacijos mazgo testavimo metu buvo atlikta kiekvieno programinės įrangos modulių testavimas. Deformacijos fiksavimo modulio testavimo metu buvo gauta 5% procentų paklaida lyginant su komerciniu įtempiu matuokliu Spider8. Deformacijų išsaugojimo modulyje buvo gautas 102.62 kb/s įrašymo greitis ir 433.85 kb/s nuskaitymo greitis, o deformacijų išsiuntimo modulyje buvo ištestuotas siuntimas MQTT protokolu ir buvo gautas 5 baitų žinutės dydis. Išorinės informacinės sistemai metu buvo atliktas funkcinis testavimas, jo metu buvo patikrintos pagrindinės sistemos funkcijos. Taip pat buvo atliktas suderinamumo testavimas, kurio metu paaiškėjo, jog sistema yra suderinama su šiuolaikinėmis

naršyklėmis ir išmaniaisiais įrenginiais. Tada prototipas buvo integruotas į "Dancerbus" elektrinį miesto autobusą. Buvo atliktas bandomasis važiavimas, specifiniu maršruto kurio ilgis buvo 4.2 kilometro ir truko 7 minutes. Per šį važiavimą buvo surinkta 218220 matavimo reikšmių. Tai reiškia, kad buvo išgautas 521Hz matavimo greitis, 16 bitų raiška. Bandomo metu autobuso kėbulas nepatyrė didelių deformacijų galinčių paveikti kompozitinį kėbulą.

LITERATŪROS SĄRAŠAS

1. Clyne, T. W.; Hull, Derek. *An introduction to composite materials*, 2019, Cambridge university press
2. Barbero, Ever J. *Introduction to composite materials design*. 2017, CRC press.
3. Lopresto, V., Caggiano, A., Teti, R. *High performance cutting of fibre reinforced plastic composite materials*. Procedia CIRP, 46, 2016, p. 71-82.
4. Agarwal, B. D.; Broutman, L. J.; Chandrashekhara, K. *Analysis and performance of fiber composites*, 2017, John Wiley & Sons.
5. Mikkelsen, L. P., Salviato, M.; Gili, J.. *From measurements errors to a new strain gauge design for composite materials*. 2015 In: 20th International Conference on Composite Materials (ICCM20). ICCM20 Secretariat.
6. Upadhyay, M. V., Panzner, T., Van Petegem, S. *Stresses and strains in cruciform samples deformed in tension*. Experimental mechanics, 2017, 57(6), 905-920.
7. Sobek, M., et al. *Carbon fiber based composites stress analysis. Experimental and computer comparative studies*. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2015. p. 95.
8. Balaji, P. S., & Karuppasamy, K. S. K. *Introduction and Application of Strain Gauges*. In Applications and Techniques for Experimental Stress Analysis 2020, pp. 57-77. IGI Global.
9. Mantič, M., Kulka, J., Faltinová, E. *Autonomous online system for evaluating steel structure durability*. Diagnostyka, 2016, 17.
10. Pawel, K., Krzysztof, G., Tomasz, D. *Position for calibrating resistance strain gauges*. In 2018 Applications of Electromagnetics in Modern Techniques and Medicine (PTZE) 2018 p. 109-112. IEEE.
11. Gad, H. E.; Gad, H. E. *Development of a new temperature data acquisition system for solar energy applications*. Renewable Energy, 2015, 74:p. 337-343.
12. Light, R. *Mosquito: server and client implementation of the MQTT protocol*. Journal of Open Source Software, 2017, 2(13): p.265.
13. Stanciu, Anca, et al. *Analysis by strain gauges of the strains in a composite material*. relation, 2010, 21(5).
14. Geng, D., Liu, Y., Shao, Z. *Delamination formation, evaluation and suppression during drilling of composite laminates: a review*. Composite Structures, 2019
15. Huang, X.; LI, Z.. *Thermal plasticity deformation buckling analysis on the carbon fiber-reinforced aluminum matrix composite material*. The International Journal of Electrical Engineering & Education, 2019, p.10.

16. Herrmann, K. M. *An Investigation of a Vertical Test Method for Large Deformation Bending of High Strain Composite Laminates*, Masters Abstracts International, 2017, 57(01), p.135
17. Gibson, R. F. *Principles of composite material mechanics.*, 2016, CRC press.
18. Šiškovski, D., Petreski, Z., Tasevski, G.. *Development of system for displacement measurement of a cantilever beam with strain gauge sensor*. Mechanical engineering scientific, 2015, p. 115.
19. Nagarajan, P. R, George, B., Kumar. *A linearizing digitizer for wheatstone bridge based signal conditioning of resistive sensors*. IEEE sensors Journal, 2017, 17(6), p.1696-1705.
20. Slanina, Z., et al. *Strain gauge sensor for the Internet of Things*. In: Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2019. International Society for Optics and Photonics, 2019. p. 1-11.
21. Li, D. S., Duan, H. W. *Strain rate and temperature effect on mechanical properties and failure of 3D needle-punched Carbon/Carbon composites under dynamic loading*. Composite Structures, 172, 2017, p. 229-241.
22. Nagy, A., Jahn, I. *Advanced Data Acquisition System for Wind Energy Applications*. Periodica Polytechnica Transportation Engineering, 2019, 47(2), p.124-130.
23. Lin, H., Zhang, Z. *A Configurable Data Acquisition System for Various Working Conditions*. In 2018 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP). IEEE, 2018, pp. 1-5
24. Silva, A. L., et al. *A study of strain and deformation measurement using the Arduino microcontroller and strain gauges devices*. Revista Brasileira de Ensino de Física, 2019, 41(3).
25. Bradney, D., et al. *A low-cost, high-speed, multi-channel Arduino-based data acquisition system for wind turbine systems*. Wind Engineering, 2019, p. 1-10.
26. Masyhur, A. H., et al. *Prototyping of Bluetooth-Based Wireless Strain Measurement System for Freight Overload Prevention*. In: 2019 6th International Conference on Electric Vehicular Technology (ICEVT). IEEE, 2019. p. 345-348.
27. Naik, N. *Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP*. In: 2017 IEEE international systems engineering symposium (ISSE). IEEE, 2017. p. 1-7.
28. Banks, A., Gupta, R. *MQTT Version 3.1. 1*. OASIS standard, 2014, 29: p. 89.
29. Kim, D. K. *Development of Mobile Cloud Applications using UML*. International Journal of Electrical and Computer Engineering, 2018, 8(1), p. 596.
30. Espressif Systems. *ESP8266EX datasheet*, [interaktyvus] Prieigą per internetą https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

31. Osinowo, M. O., Willoughby, A. A., Ewetumo, T. *Development of a Low-Cost Soil Heat Flux and Temperature Profile with Logger. Development*, AIP Conference Proceedings 1989, 2019.
32. Ali, A. S., Zanzinger, Z., Debose, D., & Stephens, B. *Open Source Building Science Sensors (OSBSS): A low-cost Arduino-based platform for long-term indoor environmental data collection*. Building and Environment, 2016, 100, p. 114-126.
33. Xu, Y., Fan, W., Cheng, P.. *Mechanical characterisation of interface shear strain of multi-layer composite pavement*. International Journal of Pavement Engineering, 2019, p. 1-7.
34. Louis, L. *Working principle of arduino and using it*. Int. J. Control. Autom. Commun. Syst, 2016, 1, p. 21-29.
35. Shin, S., Kobara, K., Chuang, C. C. *A security framework for MQTT*. In 2016 IEEE Conference on Communications and Network Security (CNS) IEEE, 2016, p. 432-436
36. Aleryani, A. *Comparative Study between Data Flow Diagram and Use Case Diagram*. International Journal of Scientific and Research Publications, 2016, 6(3), p. 124-126.
37. Barakat, S. *LAMP as a Website Development and Hosting Environment*. Research Journal of Applied Sciences, 2017, 12(5-6), p. 365-367.
38. Stępniaak, W., & Nowak, Z. *Performance analysis of SPA web systems*. In Information Systems Architecture and Technology: Proceedings of 37th International Conference on Information Systems Architecture and Technology–ISAT 2016–Part I. Springer, Cham, 2017, p. 235-247
39. Stauffer, M. *Laravel: Up & Running: A Framework for Building Modern PHP Apps*. O'Reilly Media, 2019.
40. Jailia, M., Kumar, A., Agarwal, M., & Sinha, I. *Behavior of MVC (Model View Controller) based Web Application developed in PHP and .NET framework*. In 2016 International Conference on ICT in Business Industry & Government (ICTBIG). IEEE, 2016, p. 1-5
41. Panigrahi, S. S., Shaurya, S., Das, P., Swain, A. K., & Jena, A. K. *Test Scenarios Generation Using UML Sequence Diagram*. In 2018 International Conference on Information Technology (ICIT). IEEE, 2018, p. 50-56
42. Cimbala, J. M. *Stress, strain, and strain gages*. Penn State University, 2013, 24 p.
43. Yassein, M. B., Shatnawi, M. Q., Aljwarneh, S. *Internet of Things: Survey and open issues of MQTT protocol*. In 2017 International Conference on Engineering & MIS (ICEMIS), 2017, IEEE, pp. 1-6.
44. Akintade, O. O., Yesufu, T. K., & Kehinde, L. O. *Development of an MQTT-based IoT Architecture for Energy-Efficient and Low-Cost Applications*, 2019.
45. Lakshmi, D. R., Mallika, S. S. *A review on web application testing and its current research directions*. International Journal of Electrical and Computer Engineering, 2017, 7(4), 2132

PRIEDAI

Priedas 1. Deformacijos fiksavimo mazgo kodas

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include <Adafruit_ADS1015.h>

#define SDCardPin 4

const char* ssid = "Archer";
const char* password = "*****";
const char* mqtt_server = "192.168.0.200";

File dataFile;

Adafruit_ADS1015 ads;
WiFiClient espClient;
PubSubClient client(espClient);

void setup_sd() {
  if (!SD.begin(SDCardPin)) {
    Serial.println("SD card fail!");
    while (1);
  }
  Serial.println("SD card works");
}

void setup_wifi() {
  delay(10);

  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  randomSeed(micros());

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");

    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);

    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
    }
  }
}
```

```

    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");

      delay(5000);
    }
  }
}

void setup() {
  Serial.begin(115200);

  setup_wifi();
  setup_sdcard();

  ads.setGain(GAIN_TWO);
  ads.begin();

  client.setServer(mqtt_server, 1883);
  dataFile = SD.open("data.txt", FILE_WRITE);

  Wire.setClock(400000);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  if(dataFile){
    int16_t gaugeData = ads.readADC_SingleEnded(3);

    dataFile.println(gaugeData);
    client.publish("gauge/1", String(gaugeData).c_str(), true);
  }
}

```

Priedas 2. Brokerio kodas, žinutės išsaugojimui į duomenų bazę

```
import paho.mqtt.client as mqtt
import mysql.connector as mysql
import time
import math

conn = mysql.connect(user='mqtt', password='123456', host='192.168.0.126',
database='sensor')

mqtt_topic = "gauge/#"
mqtt_broker_ip = "192.168.0.200"

client = mqtt.Client()

def on_connect(client, userdata, flags, rc):
    print "Connected!", str(rc)

    client.subscribe(mqtt_topic)

def on_message(client, userdata, msg):
    topic, sensor_no, sensor_type = msg.topic.split('/')

    if math.isnan(float(msg.payload)) == False:
        cursor = conn.cursor()

        insert_stmt = (
            "INSERT INTO measurements(sensor, type, value, created_at,
updated_at)"
            "VALUES (%s, %s, %s, %s, %s)"
        )
        data = (str(sensor_no), str(sensor_type), float(msg.payload),
time.strftime('%Y-%m-%d %H:%M:%S'), time.strftime('%Y-%m-%d %H:%M:%S'))

        try:
            cursor.execute(insert_stmt, data)

            conn.commit()
        except:
            conn.rollback()

    conn.close()

    print "Topic: ", str(msg.topic) + "\nMessage: " + str(msg.payload)

client.on_connect = on_connect
client.on_message = on_message

client.connect(mqtt_broker_ip, 1883)

client.loop_forever()
    client.disconnect()
```