

Kauno technologijos universitetas
Informatikos fakultetas

**Mikrovaldiklių panaudojimu grįstos programavimo
mokymosi aplinkos kūrimas**

Baigiamasis magistro projektas

Algirdas Barančiukas

Projekto autorius

Doc. Renata Burbaitė

Vadovė

Kaunas, 2022



Kauno technologijos universitetas

Informatikos fakultetas

**Mikrovaldiklių panaudojimu grįstos programavimo
mokymosi aplinkos kūrimas**

Baigiamasis magistro projektas

Nuotolinio mokymosi informacinės technologijos (6211BX010)

Algirdas Barančiukas

Projekto autorius

Doc. Renata Burbaitė

Vadovė

Doc. Vytenis Punys

Recenzentas

Kaunas, 2022



Kauno technologijos universitetas

Informatikos fakultetas

Algirdas Barančiukas

Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos kūrimas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektualinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Algirdas Barančiukas

Patvirtinta elektroniniu būdu

Barančiukas, Algirdas. Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos kūrimas. Baigiamasis magistro projektas / vadovė doc. Renata Burbaitė; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos inžinerija (B04), Informatikos mokslai.

Reikšminiai žodžiai: informacinės technologijos, programavimas, mikrovaldikliai, mokymosi aplinka.

Kaunas, 2022. 48 p.

Santrauka

Šiuolaikinė visuomenė nebeįsivaizduoja gyvenimo be informacinių technologijų ar sudėtingų informacinių sistemų, duomenų apdorojimo įvairia programine įranga ar informacijos paieška, robotų ar dirbtinio intelekto, techninės ir programinės įrangos naudojimo ar programavimo. Pastarasis – daugeliui iššūkis. Nors programavimo mokytis mokykloje galima ne vienerius metus, besimokantieji dažnai susiduria su įvairiomis problemomis, dėl ko mažėja mokymosi sėkmė, motyvacija. Vienas iš problemų sprendimo būdų – programavimo mokymąsi susieti su realiaame gyvenime esančiais įrenginiais, jų valdymu.

Baigiamojo projekto tikslas – sukurti ir ištirti mikrovaldiklių panaudojimu grįstą programavimo mokymosi aplinką. Šiam tikslui pasiekti buvo išanalizuota literatūra apie programavimo kalbas ir jų aplinkas, nusakytos dažniausiai su programavimo mokymu susijusios problemos bei jų sprendimo būdai.

Sociologiniame tyrime dalyvavę programavimo besimokantys mokiniai pasigedo specializuotų programavimo mokymosi aplinkų bei įvairesnių mokymosi metodų, išreiškė nuostatą, kad stokojama praktinės veiklos, susietos su realiaame gyvenime naudojamais objektais (įrenginiais).

Atsižvelgus į sociologinio tyrimo rezultatus buvo nuspręsta kurti Mikrovaldiklių panaudojimu grįstą programavimo mokymosi aplinką, todėl kurso temoms specifikuoti pritaikytas apibendrintas mokymosi scenarijaus modelis bei sukurta aplinkos darbo vietos architektūra. Pritaikius tinkamus technologinius sprendimus mokymosi aplinka buvo įdiegta, išbandyta bei naudojama mokymo(si) procese Kelmės Jono Graičiūno gimnazijoje.

Šią mokymosi aplinką vertino besimokantieji, atsakydami į jiems pateiktos anketos klausimus. Mokymosi aplinka įvertinta labai palankiai, o atsižvelgus į kai kurias pastabas bus patobulinta: sukurta daugiau pamokų scenarijų, pateikta daugiau informacijos apie mikrovaldiklius, nupirkti papildomi komponentai įtaisams konstruoti. Numatytas ir mokymosi aplinkos tęstinumas: bus naudojama kitais mokslo metais, o rajono informacinių technologijų mokytojams bus pasiūlyta išbandyti aplinką su savo mokiniais.

Barančiukas, Algirdas. Development of a Programming Learning Environment Based on the Use of Microcontrollers. Master's Final Degree Project / supervisor lect. Renata Burbaitė; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Informatics Engineering (B04), Computing.

Keywords: information technologies, programming, microcontrollers, learning environment.

Kaunas, 2022. 48 p.

Summary

Modern society cannot be imagined without information technology or complex information systems, data processing with various software or information retrieval, the use or programming of robots or artificial intelligence, hardware and software. The latter is a challenge for many. Although it is possible to study programming at school for many years, the students often face various problems that reduce their motivation for learning success. One of the ways to solve problems is to connect programming learning with real-life devices and their controlling.

The objective of the final project is to create and develop a programming learning environment based on the use of microcontrollers. Achieving the objective, the literature on programming languages and their environments was analysed, and the most common problems related to programming teaching and their solutions were established.

The students studying programming who participated in the sociological research missed specialized programming learning environments and more diverse learning methods, and expressed the attitude that there is a lack of practical activities related to the objects (devices) used in real life.

Taking into account the results of the sociological research, it was decided to develop a programming learning environment based on the use of microcontrollers, therefore a generalized learning scenario model and environment workplace architecture were applied to specify the course topics. After applying appropriate technological solutions, the learning environment was implemented, tested and used at teaching/studying process at Kelmė Jonas Graičiūnas Gymnasium.

This learning environment was evaluated by the students answering the questions in the questionnaire provided to them. The learning environment has been very well accepted, and some comments will improve it: more lesson scenarios will be developed, more information on microcontrollers will be provided, and additional components for device design will be purchased. Continuity of the learning environment is also planned: it will be used in the next school year, and region IT teachers will be suggested to test the environment with their students.

Turinys

Paveikslėlių sąrašas	8
Lentelių sąrašas	9
Įvadas.....	10
1. Programavimui mokytis skirtos aplinkos ir įrankiai.....	12
1.1. Bendrosios mokymosi aplinkų charakteristikos	12
1.2. Programavimo mokymosi aplinkos ir įrankiai.....	12
1.3. Programavimo kalbos ir jų aplinkos	13
1.4. Programavimo aplinkų naudojimas mokyklose.....	15
1.5. Programavimo mokymosi sunkumai	16
1.6. Mikropasaulių bei mikrovaldiklių programavimas.....	17
1.7. Skyriaus išvados.....	18
2. Sociologinis tyrimas siekiant išsiaiškinti mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos poreikį.....	19
Skyriaus išvados	21
3. Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos projektavimas.....	23
3.1. Reikalavimai kuriamai mokymosi aplinkai	23
3.2. Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos architektūra ir funkcionalumas	24
3.3. Kuriamos mokymosi aplinkos procesai	25
3.4. Kurso projektavimas	26
3.5. Skyriaus išvados.....	28
4. Virtualios mokymosi aplinkos galimybės kuriamoje aplinkoje	29
4.1. Virtualios mokymosi aplinkos samprata.....	29
4.2. Virtualiosios mokymosi aplinkos dalyviai ir jų poreikiai.....	30
4.3. VMA posistemiai	31
4.4. Funkciniai ir nefunkciniai VMA reikalavimai.....	31
4.5. Virtualiosios mokymosi aplinkos panaudojimo atvejų modeliai.....	32
4.6. Programinės įrangos parinkimas VMA realizuoti	35
4.7. Skyriaus išvados.....	37
5. Kuriamos aplinkos realizavimas, vertinimas.....	38
5.1. Kurso kūrimas Moodle aplinkoje ir jo dalyviai.....	38
5.2. Pamokų scenarijų pateikimai.....	39
5.3. <i>Arduino</i> programinė įranga	40
5.4. <i>Arduino</i> mikrovaldiklių su priedais naudojimas	41
5.5. Kuriamos mokymosi aplinkos vertinimas.....	42
5.6. Mokymosi aplinkos tobulinimas ir tęstinumas.....	44
5.7. Skyriaus išvados	45
Išvados	46
Literatūros sąrašas	47

Priedai.....	49
1 priedas. Problemų medis	49
2 priedas. Tyrimas „Programavimo besimokančių mokinių motyvacijos didinimo galimybės taikant praktines užduotis“	50
3 priedas. Pamokos scenarijus	53
4 priedas. Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos diegimo aktas Kelmės Jono Graičiūno gimnazijoje	60
5 priedas. Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos naudojimo tyrimas	61

Paveikslėlių sąrašas

1 pav. Mokydamasis programavimo naudojate?	20
2 pav. Kas Jus labiau motyvuotų mokytis programavimo?	21
3 pav. Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos darbo vietos architektūra	25
4 pav. Apibendrintas mokymosi scenarijaus modelis [20].....	27
5 pav. Virtualios mokymosi aplinkos pagrindiniai komponentai	30
6 pav. Administratoriaus PA diagrama administravimo posistemyje	33
7 pav. Mokytojo PA diagrama kurso kūrimo posistemyje	34
8 pav. Mokytojo ir mokinio PA diagramos turinio pateikimo posistemyje	34
9 pav. Mokytojo ir mokinio PA diagramos mokymosi proceso organizavimo ir vertinimo posistemyje	35
10 pav. Mokytojo ir mokinio PA diagramos bendravimo ir bendradarbiavimo posistemyje	35
11 pav. Sukurtas kursas VMA Moodle	38
12 pav. Vienos temos struktūra VMA Moodle kurse	39
13 pav. Programos kodas ar kodo fragmentai yra pateikiami suformatuoti	39
14 pav. <i>Arduino</i> programinės įrangos įdiegimo failo pasirinkimas.....	40
15 pav. Įdiegtos <i>Arduino</i> IDE pradinis darbo langas.....	41
16 pav. <i>Arduino UNO R3</i> rinkinys	41
17 pav. Pamokos scenarijuje numatytais užduotims atlikti pateikiami sukonstruoti įtaisai	42
18 pav. Pamokos scenarijuje pateiktų užduočių vertinimas	43

Lentelių sąrašas

1 lentelė. Programavimo kalbų palyginimas (Mannila ir de Raadt, 2006).....	14
2 lentelė. Programavimo kalbų reitingas (tiobe.com, 2022 m. balandis).....	15
3 lentelė. Reikalavimai kuriamai mokymosi aplinkai	24
4 lentelė. Programavimo pradmenų modulyje mokinių įgyjami įgūdžiai, žinios ir supratimas	26
5 lentelė. Posistemių funkciniai ir nefunkciniai reikalavimai	32
6 lentelė. VMA palyginimas.....	36

Ivadas

Informatikos mokslas evoliucionuoja labai greitai, jo turinį ir ribas darosi vis sunkiau griežtai apibrėžti. Mokykloje dėstomos informacinės technologijos apima labai platų kompiuterių panaudojimo spektrą. Pagal informacinių technologijų bendrąsias pagrindinio ir vidurinio ugdymo programas mokyklose mokomasi kompiuterių ir algoritminių procesų, techninės bei programinės įrangos klasifikacijos bei praktinio taikymo ir poveikio visuomenei. Mokykloje dėstomas informacinių technologijų kursas apima nedidelę dalį informatikos mokslo sričių. Informatikos mokslas apima: informacijos paiešką, logiką, techninės ir programinės įrangos projektavimą, programavimo paradigmas, programavimą ir programavimo kalbas, duomenų bazių kūrimą, kompiuterinę grafiką, kompiuterių tinklus, dirbtinį intelektą, modelius ir jų transformacijas, informacinių technologijų taikymą ir informacinių sistemų kūrimą. Taip pat daug dėmesio skiriama saugumo ir privatumo internete aspektams bei autorių teisių apsaugai. Informatika glaudžiai susijusi su intelektualiomis sistemomis, kompiuterine rega, bioinformatika, robotikos pagrindais.

Šiuolaikinė visuomenė su kompiuterinėmis sistemomis susiduria kasdien ir beveik kiekviename žingsnyje. Kompiuteriai ar jų sistemos naudojamos transporte, sveikatos ir farmacijos sferose, įvairiose pramonės šakose. Informatikos pagrindus mokykloje gavę mokiniai gyvenime bus ne tik gerais technologijų naudotojais, bet ir imsis įvairių inovacijų savo ar visos visuomenės gyvenimo kokybei gerinti.

Įvairių profesijų atstovų darbas jau neįsivaizduojamas be kompiuterių ar jų sistemų naudojimo. Todėl visi dirbantieji, būdami geri savo srities profesionalai, turi mokėti tinkamai naudotis šiuolaikinėmis priemonėmis.

Per informacinių technologijų pamokas įgytus loginio ir algoritminio mąstymo įgūdžius mokiniai taiko ir taikys įvairiose gyvenimo srityse nuo inžinerijos ir gamtos bei tikslųjų mokslų, iki humanitarinių ir socialinių mokslų.

Remiantis išvardintomis priežastimis galima teigti, kad mokiniai per informatikos pamokas turi tapti ne tik profesionaliais kompiuterių kaip darbo įrankio naudotojais, bet ir suprastų kompiuterių ir išmaniųjų įrenginių darbo principus, mokytūsi projektuoti ir kurti programinę įrangą.

Šiame darbe apsisistota plačiau ties viena sritimi – programavimo mokymu mokykloje.

Išmokti programavimo daugeliui yra didelis iššūkis. Programavimo mokytis tradiciškai pradedama nuo žinių (susipažįstama su sintakse ir semantika) mažesnę dėmesį skiriant problemų sprendimo strategijoms. Winslow'as [1] teigė, kad pradedantieji programuotojai išmano atskirų teiginių sintaksę ir semantiką, tačiau nežino, kaip šias funkcijas sujungti į galiojančias programas, nors programų rašymas apima keletą etapų – programos kodo rašymo, testavimo, derinimo ir paleidimo. Šie etapai gali būti sudėtingi ne vienam besimokančiajam, todėl mažėja jų motyvacija. Kai kurie mokslininkai besimokantiems motyvuoti siūlo stiprinti mokinių ir mokytojų sąveiką, taikyti vizualizavimo, žaidimų, galvosūkių ar skaičiavimų metodus. Labai svarbu išvystyti gerą besimokančiųjų algoritminį mąstymą, apimančių problemų sprendimą, sistemų projektavimą ir žmogaus elgesio supratimą.

Nors Lietuvoje susiformavusios gilios programavimo mokymo(si) tradicijos, tačiau modernizuojant mokymąsi programuoti daugiau dėmesio turi būti skiriama realaus pasaulio uždaviniams spręsti koncentruojantis į problemų sprendimą ir informatinio mąstymo ugdymą [2].

Darbo problema. Nepakankama mokinių motyvacija mokytis programavimo dėl negebėjimo apjungti atskiras programos dalis ar funkcijas į veikiančias programas.

Darbo objektas. Darbe tiriama mikrovaldiklių panaudojimu grįsta programavimo mokymosi aplinka.

Darbo tikslas. Sukurti ir ištirti mikrovaldiklių panaudojimu grįstą programavimo mokymosi aplinką, padedančią gerinti programavimo žinias ir ugdyti programavimo bei problemų sprendimo įgūdžius.

Darbo uždaviniai:

1. atlikti programavimui mokytis skirtų aplinkų analizę;
2. pagrįsti mikrovaldiklių panaudojimo programavimo mokymosi aplinkoje aktualumą;
3. sukurti mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos modelį;
4. įgyvendinti sukurtą modelį Kelmės Jono Graičiūno gimnazijoje įdiegiant mikrovaldiklių panaudojimu grįstą programavimo mokymosi aplinką;
5. ištirti sukurtą aplinką ir remiantis gautais tyrimo rezultatais ją tobulinti.

Darbo rezultatas. Sukurta mikrovaldiklių panaudojimu grįsta programavimo mokymosi aplinka, kuri didina mokinių motyvaciją mokytis programavimo, gerina programavimo žinias, ugdo programavimo ir problemų sprendimo įgūdžius. Mikrovaldiklių panaudojimu grįsta programavimo mokymosi aplinka įdiegta Kelmės Jono Graičiūno gimnazijoje (žr. 4 priedą), išbandyta bei ištirta.

1. Programavimui mokytis skirtos aplinkos ir įrankiai

1.1. Bendrosios mokymosi aplinkų charakteristikos

Norint tinkamai suprojektuoti ir įdiegti programavimui mokytis skirtas aplinkas labai svarbu atsižvelgti į besimokančiųjų mokymosi stilių, žinių ir gebėjimų lygį, įvardinti mokymosi sunkumus ir nustatyti jų priežastis. Dažniausiai besimokantieji susiduria su uždavinio sprendimo algoritmo sudarymo ar esamo algoritmo modifikavimo, programavimo technikų panaudojimo sunkumais. Taip pat trūksta problemų sprendimo įgūdžių.

Gomes ir Mendes [3] išskyrė svarbiausias charakteristikas, apibūdinančias efektyvią mokymosi aplinką:

- 1) nuolat stebėti kiekvieno besimokančiojo žinių lygį, bei nustatyti dominuojantį mokymosi stilių – tai efektyvesnio ir daugiau individualizuoto mokymosi galimybė;
- 2) naudoti programavimui mokytis modelius (uždavinius skaidyti į savarankiškas dalis, kada atsiranda galimybė pritaikyti žinias panašioms užduotims atlikti), kurie ugdo problemų sprendimo ir kritinio mąstymo gebėjimus;
- 3) nemažą dėmesį skirti žaidimams – jų dėka galima didinti mokinių mokymosi motyvaciją, ugdyti tinkamus problemų sprendimo įgūdžius;
- 4) naudoti įvairius algoritmų kūrimo įrankius ir techninius išteklius, kurių dėka besimokantys įgyja algoritmų kūrimo bei jų pritaikymo praktikoje žinių bei įgūdžių.

1.2. Programavimo mokymosi aplinkos ir įrankiai

Kelleher ir Pausch [4] programavimo aplinkas skirsto į dvi grupes – mokymo ir įgalinančiąsias sistemas:

Mokymo sistemos (angl. *Teaching Systems*) yra skirtos norintiems išmokti programuoti. Šiose sistemose įdiegti įrankiai, kurie pasižymi panašiomis funkcijomis kaip ir universalios programavimo kalbos (angl. *general-purpose languages*) ir išryškina svarbiausius programavimo proceso aspektus: komandų pateikimą kompiuteriui ir jo atliekamų veiksmų supratimą. Mokymo sistemos pasižymi tuo, kad: 1) supaprastinama programavimo kalba, 2) kalba pritaikoma specifinių sričių uždaviniams spręsti, 3) įvedamas automatinis sintaksės klaidų taisymas, 4) surandami alternatyvūs programų kodo pateikimo būdai: programų kodo konstravimas iš grafinių ar fizinių objektų, programų kūrimas naudojant sąsajos veiksmus (pvz., mygtukų paspaudimus, judėjimą erdvėje), programų kūrimas taikant daugelį mechanizmų (programų specifikavimą ir pateikimą įvairiais formatais), 5) programų struktūrizavimas, panaudojant esamus ir kuriamus naujus programavimo modelius, 6) supratimas apie programų veikimą didinamas stebint programų vykdymą, naudojant mikropasaulių galimybes [9] ir kuriant programų vykdymo modelius.

Įgalinančiųjų sistemų (angl. *Empowering Systems*) dėka besimokantieji gali suprojektuoti ir įgyvendinti jiems reikalingas programas. Šiose sistemose daugiausia dėmesio skiriama programavimo kalbų ir metodų kūrimui. Kodo sudėtingumo problemos sprendžiamos įvairiais būdais: 1) veiksmams ir sąlygoms nurodomi sąsajoje nekeičiant programos kodo, 2) tobulinant programavimo kalbas ir jų sąsajas, didinant

kalbų suprantamumą. Įgalinančiose sistemose besimokantiejiems sudaromos sąlygos kurti ir tyrinėti įvairių pažinimo sričių modelius, pritaikant universaliąsias programavimo kalbas specifinėms sritims.

Akcentuodamas, kad įvadiniuose programavimo kursuose dažniausiai mokoma sintaksės ir semantikos, o ne problemos sprendimo, Law'as [5] pristatė e. mokymosi sistemą *PASS* (programavimo užduočių vertinimo sistema), kuri gerokai palengvina pradedančiųjų programuotojų mokymąsi. Programoje *PASS* instruktorius įkelia programavimo problemą kartu su bandymų atvejų rinkiniu. Sistema automatiškai vertina sprendimą pagal dėstytojo pateiktus testo atvejus ir suteikia studentui greitą grįžtamąjį ryšį. Todėl *PASS* pradedantiejiems taiko palaipsniui atspindintį požiūrį į mokymąsi programuoti ir teikia studentams savalaikį bei aktualų grįžtamąjį ryšį, kad palengvintų jų mokymosi procesą.

Įvado į programavimą aplinka gali būti ir išmanieji telefonai bei programavimas jais. Reardon [6] teigia, kad studentų mokymasis, naudojamas kartu su programų kūrimu išmaniaisiais telefonais, yra veiksminga įvadinio programavimo kurso pedagogika.

Kaip viena iš programavimo mokymo aplinkų buvo siūlomas ir nesudėtingų žaidimų programų tyrinėjimas [7], o besimokančiųjų rodikliai buvo geresni nei klasėse, kuriose buvo taikomi tradiciniai programavimo mokymo(si) metodai.

Kadangi mikrovaldikliai plačiai naudojami įvairiuose įtaisuose, programavimo pradmenų mokymuisi sėkmingai gali būti naudojamos mikrovaldikliais grįstos mokymosi aplinkos.

1.3. Programavimo kalbos ir jų aplinkos

Dagienė ir Urbonienė [8] pastebi, kad programavimo kalbos dažnai būna integruotos su kompiliatoriumi, taip sukuriant aplinką. Todėl dažnai kalbant apie programavimo kalbas jos tapatinamos su aplinkomis.

Programavimo mokymas glaudžiai siejamas su programavimo kalbos aplinka. Dažnai programavimo mokymosi sėkmė priklauso nuo tinkamai pasirinktos programavimo kalbos aplinkos. Kurį aplinką pasirinkti lemia įvairūs kriterijai. Daug mokslininkų tyrinėjo programavimo kalbų ir jų aplinkų tinkamumo mokytis kriterijus. Mokslininkai išskyrė tokius kriterijus, kaip: reikalavimas aprašyti kintamuosius ir tipus, tinkamumas mokytis keliais lygiais, sintaksės paprastumas ir aiškumas, galimybė skaidyti programą į savarankiškas dalis, patikimumas, saugumas, efektyvumas, pakankamas mokymosi medžiagos kiekis, lengvai išmokstama bei nesunku pereiti prie kitos programavimo kalbos, nemokama. Šis sąrašas nėra baigtinis. Manilla ir de Raadt [9] (2006) atliko įvairius tyrimus ir suformulavo 17 kriterijų, skirtų įvertinti programavimo kalbų tinkamumą mokytis. Kriterijai buvo suskirstyti į keturias grupes: mokymasis, dizainas ir aplinka, palaikymas ir prieinamumas, nereikalingos pradinės programavimo žinios. Mokslininkai pirmiausia pagal šiuos kriterijus paprašė įvertinti savo sukurtas programavimo kalbas *LOGO* kūrėjo S. Papert, *Pascal* kūrėjo N. Wirth, *Python* kūrėjo G. van Rossum ir *Eiffel* kūrėjo B. Meyer. Vėliau buvo įvertintos ir kitos mokymui skirtos programavimo kalbos (1 lentelė).

1 lentelė. Programavimo kalbų palyginimas (Mannila ir de Raadt, 2006)

Kriterijai	Programavimo kalbos								
	C	C++	Eiffel	Java	JavaScript	Logo	Pascal	Python	VisualBasic
Mokymasis									
1. Tinka mokyti įvairiais lygiais	-	-	+	-	-	+	+	+	-
2. Tinka įvairiems įtaisams valdyti	-	-	+	+	+	+	-	+	+
3. Siūlo vientisą programų kūrimo sistemą	+	+	+	+	+	-	+	+	+
4. Skatina programinės įrangos mokymąsi	-	-	+	±	-	+	-	-	-
Dizainas ir aplinka									
5. Interaktyvi ir palengvina greitą kodo kūrimą	-	-	-	-	-	+	-	+	-
6. Skatina rašyti iš karto teisingas programas	-	±	+	±	-	-	-	±	-
7. Leidžia programas skaidyti į savarankiškas dalis	+	+	+	+	+	+	+	+	+
8. Vientisa programų kūrimo aplinka	-	-	+	±	-	-	-	-	-
Palaikymas ir prieinamumas									
9. Nemaža kalbos naudotojų bendruomenė	+	+	+	+	+	-	-	+	+
10. Atvirojo kodo, todėl kiekvienas gali prisidėti prie jos kūrimo	-	-	-	-	-	-	-	+	-
11. Nuolat palaikoma įvairiose aplinkose	+	+	+	+	+	+	+	+	-
12. Laisvai ir lengvai prieinama	+	+	+	+	+	+	+	+	-
13. Yra pakankamai geros mokymosi medžiagos	-	+	+	+	-	+	+	+	+
Nereikalingos pradinės programavimo žinios									
14. Naudojama ne tik švietime	+	+	+	+	+	-	-	+	+
15. Lengvai išplečiama	+	+	+	+	-	-	-	+	+
16. Patikima ir efektyvi	+	+	+	+	+	-	+	+	+
17. Programos kuriamos ne tik rašant tekstą	-	+	+	+	+	+	-	+	+
Iš viso:	8	11	15	14	9	9	7	15	9

Apžvelgiant pateiktus rezultatus galima teigti, kad geriausios mokymui skirtos programavimo kalbos yra *Eiffel* ir *Python*. *Eiffel* programavimo kalba tenkina visus mokymosi grupės kriterijus (*Python* neskaitina programinės įrangos mokymosi), yra vientisa programų kūrimo aplinka. Tuo tarpu *Python* programavimo kalba skiriasi nuo *Eiffel*: ji yra atviro kodo, todėl kiekvienas gali prisidėti prie jos kūrimo, vystymo bei yra interaktyvi ir palengvina greitą kodo kūrimą. Programavimo kalba yra neatsiejama nuo jos aplinkos, todėl *Eiffel*, kuri pas mus Lietuvoje nėra populiarė, turi pranašumo prieš *Python*.

Žvelgiant į visas 1 lentelėje pateiktas programavimo kalbas galima teigti, kad programavimui mokytis tinkamiausios yra *Eiffel*, *Python*, *Java*, *C++*. Pastaroji programavimo kalba yra populiarė Lietuvos mokyklose, o aktyvios IT bendruomenės narių dėka kelią į Lietuvos mokyklas skinasi ir *Python*.

Tiobe.com [10] žiniatinklyje skelbimas TIOBE programavimo bendruomenės indeksas yra programavimo kalbų populiarumo rodiklis. Žvelgiant į 2022 metų balandžio mėnesio penketuką (2 lentelė) matyti, kad pirmoje vietoje tvirtai laikosi *Python* programavimo kalba, o *C++* yra ketvirtojoje vietoje.

2 lentelė. Programavimo kalbų reitingas (tiobe.com, 2022 m. balandis)

Vieta 2022 m. balandis	Vieta 2021 balandis	Programavimo kalba	Reitingas 2022 m. balandis	Pokytis lyginant su 2021 m. balandžiu
1	3	<i>Python</i>	13.92%	+2.88%
2	1	<i>C</i>	12.71%	-1.61%
3	2	<i>Java</i>	10.82%	-0.41%
4	4	<i>C++</i>	8.28%	+1.14%
5	5	<i>C#</i>	6.82%	+1.91%

Žvelgiant į programavimo kalbų populiarumo rodiklių dvidešimtmetį matyti *Python* programavimo kalbos populiarėjimas: 12-oji vieta (2002 m.), 8-oji vieta (2012 m.) ir 1-oji vieta (2022 m.). Tuo tarpu stabiliai 3-oje vietoje buvusi *C++* programavimo kalba (2002 ir 2012 m.) smuktelėjo į 4-ąją vietą, o didesnę nuosmukį patyrė *Java* – nuo 1-osios vietos (2002 ir 2012 m.) iki 3-osios (2022 m.).

Python programavimo kalbos populiarumą greičiausiai lėmė jos paprastumas: mažiau įvairių sąvokų, atlaidžiau žiūrima į kai kurias programavimo klaidas, programos tekstas konstruojamas iš atskirų dalių, nėra griežtumo. Kai kuriuose šalyse *Python* rekomenduojama dėstyti pirmiausia. Programavimo kalbos populiarumą lemia ir jos praktinis pritaikymas: nuo tinklalapių ar duomenų bazių kūrimo iki dirbtinio intelekto, taip pat galima kurti žaidimus, valdyti įtaisus.

1.4. Programavimo aplinkų naudojimas mokyklose

Pagal dabar galiojančias pagrindinio ugdymo bendrąsias programas: informacinės technologijos [11], patvirtintas 2008 metais, programavimo pradmenų mokymui siūlomos naudoti „programavimo aplinkos (pvz., *Komenskio Logo, Imagine Logo, FreePascal*)“. 10-oje klasėje programavimo pradmenų modulis nėra privalomas, jį mokosi tik pasirinkusieji. Tiesa, *FreePascal* kalbos daugelis mokytojų jau seniai atsisakė, vietoje jos pasiūlę *C++*. Pastaraisiais metais vis labiau pradeda kalbėti apie *Python* programavimo kalbos mokymą mokykloje.

Vidurinio ugdymo bendrosiose programose: informacinės technologijos [12], patvirtintuose 2011-aisiais metais, programavimas yra kaip vienas iš trijų siūlomų išplėstinio kurso modulių, kurį mokyti renkasi mokiniai. Programose numatyta, kad „pasirenkamųjų modulių ugdymo turinys parengtas taip, kad nė vienas iš jų nėra susietas su konkrečia programine įranga. Mokytojas laisvas pasirinkti programinę įrangą atsižvelgdamas į savo ir mokinių pasirėngimą, mokinių pageidavimą, mokyklos galimybes ir pan.“. Tačiau pasirinkimo laisvė yra apribota informacinių technologijų brandos egzamino programa, kurioje nurodyti reikalavimai programinei įrangai egzamino metu. Dažniausiai pasirenkamąjį programavimo modulį renkasi mokiniai, kurie savo ateitį žada sieti su kompiuteriais, inžinerija, informacinėmis technologijomis, todėl jiems reikalingas IT brandos egzaminas. Norėdami sėkmingai išlaikyti šį egzaminą, mokiniai programavimo mokymuisi turi pasirinkti vieną iš egzamino programoje nurodytų aplinkų: „*FreePascal, Code::Blocks, DEV C++* ir *Visual Studio Code Python LT*“ [13]. *Python* programavimo kalbą IT brandos egzamine pirmą kartą leista naudoti 2021-aisiais metais.

Kiekvienais mokslo metais šalies mokyklose organizuojamos mokomųjų dalykų olimpiados ar įvairūs konkursai, kuriuose dalyvauja gabūs mokiniai. Kiekvienais metais organizuojama ir Lietuvos mokinių informatikos olimpiada, kuri vyksta keliais etapais. Mokykliniame ir rajoniniame olimpiados etapuose atliktoms užduotims pateikti programavimo kalbas parenka atitinkamų institucijų (mokyklos ar rajono švietimo įstaigos, atsakingos už dalykines olimpiadas, vadovo) sudarytos komisijos. Baigiamajame etape mokiniams leidžiama programuoti C/C++ kalbomis.

Apibendrinant galima teigti, kad pasirinkimo laisvė renkantis programavimo kalbas mokykloje atrodo yra didelė, tačiau mokiniai, norintys sėkmingai išlaikyti IT brandos egzaminą bei dalyvauti olimpiadose, turi mokytis C++ programavimo kalbos.

1.5. Programavimo mokymosi sunkumai

Programavimas – sudėtingas intelektinis procesas [8], todėl norint to išmokti ne vienam kyla įvairių iššūkių. Sudominti besimokančiuosius programavimu galima pateikiant jį labai paprastai, aiškiai, parenkant tinkamas aplinkas bei užduotis. Taip pat reiktų atkreipti ypatingą dėmesį į tai, kad sėkmė lydės ne tuos, kurie išmoks gerai kalbos sintaksę bei semantiką, kurie bus perskaitę daugybę įvairių teisingai parašytų programos kodų, bet tuos, kurie daug dirbs savarankiškai, kad suformuotų tinkamus įgūdžius, išmoktų logiškai mąstyti.

Problemas, kurios atsiranda mokant programavimo, galima suskirstyti į 4 grupes: pedagoginės, pažinimo, technologinės, mokymosi turinio [14] (žr. 1 priedą).

Pedagoginės problemos. Dažnai susiduriama su problema, kad mokymo modeliai neatliepia mokinių mokymosi poreikių. Besimokantiesiems įdomu būtų programuoti įtaisus, valdyti juos, tuo tarpu daugiau koncentruojamasi į sintaksę bei semantiką ar šabloninių programų rašymą, kurios neverčia mokinio kritiškai mąstyti, įveikti pasitaikančias problemas. Neretai būna, kad mokymo metodai neatitinka mokinių mokymosi stilių, per daug teorinių žinių ir per mažai praktikos, kurios metu formuojami geri programavimo įgūdžiai bei loginis mąstymas. Programavimo mokymas dažnai negali būti individualizuojamas, nes klasėse labai daug mokinių. Reiktų nepamiršti ir mokinių mokymosi motyvacijos, kuri dažnai būna nepakankama.

Pažinimo problemos. Mokantis programuoti operuojama aukšto lygio abstrakcijomis, todėl besimokantiesiems dažnai sunku viską teisingai suprasti bei susieti tai su programų rašymu, kas programavimo mokymesi užima be galo svarbų vaidmenį. Teorinės žinios apie programavimą visada turi būti glaudžiai siejamos su praktine mokinių veikla: tik daug savarankiškai dirbantys besimokantieji gali pasiekti geresnių rezultatų programavime. Neretai mokant programavimo mokiniams yra pateikiamos parašytos programos, kurias siūloma analizuoti ir iš jų mokytis. Tačiau yra didžiulis skirtumas tarp programų supratimo ir iš anksto parašytų programų modelių.

Technologinės problemos. Geresnių rezultatų programavime padeda pasiekti tinkamos kalbos bei aplinkos parinkimas ir mokymo formatas. Interaktyvus programavimo mokymasis gali patirti nesėkmę dėl pažinimo ir pedagoginių problemų. Taip pat reiktų nepamiršti, jog tradicinės mokymosi valdymo sistemos neapima visų programavimo mokymosi kontekstų.

Mokymosi turinio problemos. Programavimas yra dinamiškas, tačiau mokymosi medžiaga dažnai pateikiama statiniu formatu, dėl ko tampa ne tokia patraukli besimokančiajam. Pasigendama adaptuoto turinio pagal kiekvieno mokinio poreikius. Pateikiant mokymosi turinį labai svarbų vaidmenį vaidina jo vizualizacija.

1.6. Mikropasaulių bei mikrovaldiklių programavimas

Daugelis patyrusių mokytojų programavimo įvadui renkasi mikropasaulius, kurie leidžia besimokantiesiems greitai ir intuityviai išmokti abstrakčių programavimo sąvokų bei suformuoti tinkamus praktinius programavimo įgūdžius. Tarp šių abiejų dalių turi būti neatsiejama sąveika.

Djelil'as bei kiti mokslininkai [15] sakė, kad įvadiniam programavimui mokytis dažnai naudojami mikropasauliai, nes jie besimokantiesiems yra labai patrauklūs dėl įtraukimo į mokymosi procesą ir greito grįžtamojo ryšio.

Mikropasaulių (mažų pasaulių) sąvoką išpopuliarino Papert'as [16], savo sukurtoje *LOGO* programavimo kalboje apibrėžęs komandų rinkinį mažame pasaulyje judančiam vėžliukui valdyti.

Kalbant apie kompiuterinius mikropasaulius, juos galima apibūdinti kaip mažus dirbtinius pasaulius, neatsiejamus nuo realaus pasaulio. Kitaip sakant realiame pasaulyje besimokantiesiems gerai žinomi objektai susieti su formaliame (abstrakčiame) pasaulyje esančiais mokslo objektais.

Mikropasaulį galima apibrėžti kaip skaitmeninę aplinką, kurioje daug interaktyvumo, tačiau išlaikomas šios aplinkos paprastumas bei patrauklumas ir sukuriamas greitas grįžtamasis ryšys. Daug kam gali pasirodyti, kad mikropasaulių programavimas yra žaidimų pasaulis, tačiau iš tikro tai nėra žaidimai.

Mikropasaulių programavimas paremtas metaforomis, kurios abstrakčias programavimo sąvokas priartina prie realiame pasaulyje naudojamų objektų, dėl ko besimokantiesiems daug lengviau viską susieti bei suprasti.

Ne mažiau svarbūs mikropasaulių programavime vizualizacija ir interaktyvumas, kurie, anot Henriksen'o ir Kölling'o [17], mokymosi aplinkoje leidžia eksperimentuoti, didina besimokančiųjų smalsumą ir įsitraukimą.

Mikropasauliams ne svetimas ir konstruktyvizmas: besimokantieji savo žinias ir gebėjimus gali konstruoti per patirtį, eksperimentuodami, atlikdami įvairias užduotis individualiai, savo tempu. Greitas grįžtamasis ryšys parodo padarytas klaidas, kurias mokiniai iškart gali taisyti. Tai didina mokinių motyvaciją.

Dar vienas mikropasaulių programavimo bruožas yra žaidimai, kurie gali būti naudojami motyvacijos stiprinimui, pasitaikančių problemų sprendimų paieškai, naujų žinių ir gebėjimų formavimui. Per žaidimus besimokantieji mokosi konstruktyvizmo, kuris yra mikropasaulių programavimo pagrindinė sudedamoji dalis.

Mikropasaulių programavimą galima keisti mikrovaldiklių programavimu. Kalbant apie pastarąjį procesą, jo metu yra programuojami realiame pasaulyje naudojami įtaisai, kuriuose daug interaktyvumo.

Mokymasis programuoti mikrovaldiklius paremtas konstruktyvizmu kaip ir mikropasaulių programavimas, jame galima rasti žaidybinimo elementų, kurie iš tikro nėra žaidimai. Vienas iš labiausiai naudojamų mikrovaldiklių – *Arduino*. Tai atvirojo kodo lengvai naudojama aparatinė ir programinė įranga. *Arduino* integruotoje kūrimo aplinkoje galima rašyti programas ir kelti jas į mikrovaldiklio plokštę be didesnių pastangų. Konstruojami įtaisai su mikrovaldikliais, į kuriuos įkeltas sėkmingai parašytas programos kodas, gali dirbti atskirai arba prijungti prie kompiuterio, kuriame buvo rašyta programa.

Kalbėdamas apie *Arduino*, Galadima's [18] išskiria ***techninę įrangą***: mikrovaldiklio plokštė ir jos techniniai parametrai, maitinimas per USB kabelį ar išoriniu šaltiniu, įvesties ir išvesties galimybės, ryšys su kompiuteriu; ***programinę įrangą***: atviro kodo programa, suderinta su įvairiomis kompiuterių operacinėmis sistemomis; ***aplinką***: yra teksto redaktorius programos kodui rašyti, įrankių juosta, įvairios funkcijos. Visi šie komponentai yra svarbūs rašomų programų funkcionalumui bei įtaisų su mikrovaldikliais mobilumui.

Šiandien gyvename technologijų pilname pasaulyje, o beveik kiekviename butyje naudojame įrenginyje yra mikrovaldiklis. *Arduino* mikrovaldiklis gali būti naudojamas įvairiuose išsilavinimo lygiuose – nuo šviestukų mirksėjimo iki sudėtingų įtaisų valdymo, kas skatina besimokančiuosius domėtis daiktų su mikrovaldikliais projektavimu bei kūrimu. Įvairių projektų metu atliekant užduotis yra gerinamas kritinis mąstymas, problemų sprendimo įgūdžiai.

1.7. Skyriaus išvados

1. Šiuolaikinės programavimo kalbos yra sudėtinė mokymosi aplinkų dalis, nuo kurių tinkamo pasirinkimo dažnai priklauso mokymosi sėkmė. Lietuvos mokyklose dominuoja C++ programavimo kalba, kurios pasirinkimą lėmė tai, kad ši kalba įtraukta į informacinių technologijų valstybinio brandos egzamino programą ir naudojama mokinių informatikos olimpiadose.
2. Mokantis programavimo gali atsirasti pedagoginių, pažinimo, technologinių ar mokymosi turinio problemų, todėl labai svarbi yra teorijos ir praktikos sintezė: sėkmė lydės tuos besimokančiuosius, kurie ne tik gerai išmoks programavimo kalbos sintaksę bei semantiką, bet ir daug dirbs savarankiškai, suformuos gerus praktinius bei ugdysis ir gilins loginio, analitinio ir kritinio mąstymo įgūdžius.
3. Programavimui mokytis geriausia naudoti tas aplinkas, kurios skatina ieškoti problemų sprendimų, įtraukia besimokantį į mokymosi procesą bei suteikia greitą grįžtamąjį ryšį, leidžia eksperimentuoti ir per pažinimą susieti abstrakčius programavimo objektus su realiame gyvenime esančiais objektais.

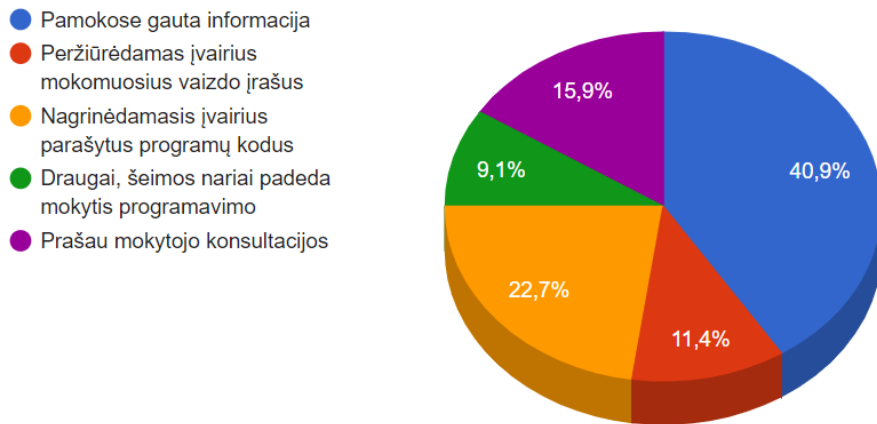
2. Sociologinis tyrimas siekiant išsiaiškinti mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos poreikį

Pagal pagrindinio ugdymo bendrąsias programas mokiniai mokytis programavimo gali pasirinktinai 9-10 klasėje (1 pamoka per savaitę 9 arba 10 klasėje), o pagal vidurinio ugdymo bendrąsias programas – pasirinkus „Programavimo“ modulį, kai 11 ir 12 klasėse skiriama po 1 pamoką per savaitę, arba 2 pamokos per savaitę tik 11 arba tik 12 klasėje (mokykloms suteikta teisė pasirinkti, kaip bus mokomasi). Tokio pamokų skaičiaus nepakanka mokiniams gerai įsisavinti šį nelengvą mokomąjį dalyką, o kalbėti apie puikius susiformavusius programavimo įgūdžius negali būti ir kalbos. Nuolatinis programų rašymas papildo teorines žinias ir stiprina praktinius įgūdžius, jos yra taikomos praktiškai, tačiau dažniausiai kyla problemų ir mokiniams reikia vienokios ar kitokios pagalbos rezultatui pasiekti. Keletas nesėkmingų bandymų kartais mokinius nuvilia, krinta jų motyvacija mokytis programavimo, atsiranda spragos. Dėl tos priežasties krinta mokinių pasiekimai, žinios, įvertinimai.

Siekiant išanalizuoti Kelmės Jono Graičiūno gimnazijos programavimo besimokančių mokinių nuomonę apie šio dalyko mokymą(-ąsi), motyvacijos didinimą bei įgytų žinių praktinį pritaikymą buvo atliktas sociologinis tyrimas (žr. 2 priedą). Buvo apklausta 18 Kelmės Jono Graičiūno gimnazijos III (66,7 %) ir IV (33,3 %) klasių mokinių, iš kurių 88,9 % buvo vaikinai ir 11,1 % – merginos. Didžioji apklausoje dalyvavusių respondentų dalis (61,1 %) programavimo mokosi antrus metus, 27,8 % – trečius metus, likusieji tik pirmus ar ketvirtus metus (abu po 10 %). Lygiai pusė apklaustųjų mokytis programavimo pasirinko todėl, kad jiems patinka kompiuterių mokslas, trečdalis mokinių šio dalyko mokytis rinkosi kartu su draugais, norėdami susipažinti su programavimu. Šiek tiek daugiau nei dešimtdalis (11,1 %) gimnazistų savo ateitį sieja su programavimu – nori būti programuotojais, likusiems (5,6 %) programavimo mokytis pasiūlė mokytojas.

Didžioji dalis gimnazistų (85,7 %) programavimo mokosi C++ programavimo kalba, 9,5 % - *Python*, likę – *JavaScript*. Kelmės Jono Graičiūno gimnazijoje mokiniams, kurie mokosi programavimo III–IV klasėse, tam skiriamos 2 savaitinės valandos – dvigubai daugiau, negu numatyta bendrojo ugdymo programose. Papildomos valandos skiriamos gimnazijos administracijos sprendimu iš rezervinių valandų. 55,6 % respondentų teigia, jog šių pamokų pakanka, tačiau daugiau nei ketvirtadalis apklaustųjų (27,8 %) sako, jog kursas platus, todėl jam gerai įsisavinti reikėtų skirti daugiau pamokų. 16,7 % mokinių šiuo klausimu nuomonės neturi.

Net 83,3 % gimnazistų teigiamai atsiliepia apie pamokas, kuriose mokomasi programavimo ir teigia, kad jų metu gauna daug ir įvairios informacijos. Likę respondentai dar mokosi savarankiškai. Mokydamiesi mokiniai naudoja įvairius šaltinius (1 pav.)



1 pav. Mokydamasis programavimo naudojate?

Prie sėkmingo programavimo mokymosi prisideda ir mokykloje kompiuterių klasėse esanti techninė įranga. Daugiau kaip pusė (51,4 %) gimnazistų mano, jog klasėje esančių kompiuterių charakteristikos įgalina be problemų mokytis programavimo (naudojama aplinka *Code::Blocks* patenkina besimokančiųjų poreikius). 37,1 % respondentų namuose turi kompiuterį su įdiegta ir puikiai veikiančia aplinka, tačiau dalis mokinių (8,6 %), kurie naudoja *online* programavimo aplinką, kartais susiduria su sunkumais.

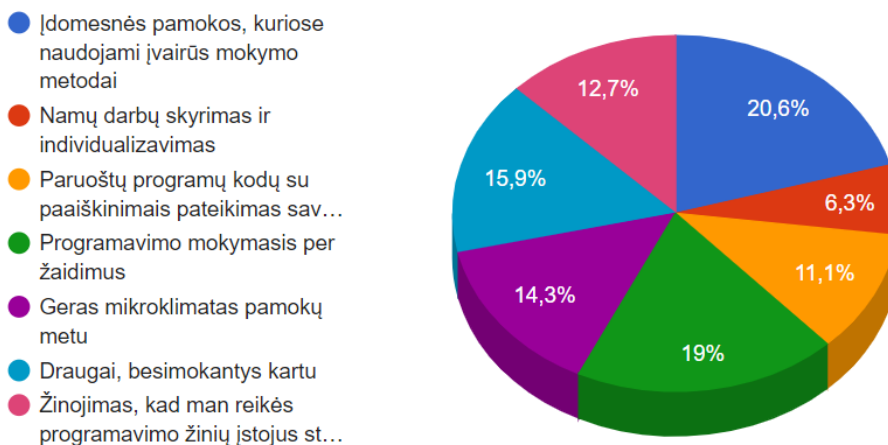
83,3 % apklausoje dalyvavusių respondentų teigia, kad pakanka mokomųjų programų ar mokymuisi skirtų aplinkų, tačiau 16,7 % respondentų teigia, kad mokykloje nenaudojamos jokios adaptuotos mokyklai programavimo mokymo(si) aplinkos.

Daugiau nei ketvirtadalis apklaustųjų (27,8 %) norėtų mokytis daugiau nei vieną programavimo kalbą lygiagrečiai. Tiek pat mokinių to daryti nenorėtų, o likę neturi nuomonės šiuo klausimu.

Lygiai pusė apklausoje dalyvavusių respondentų teigia, kad programavimo modulio programa mokykloje atitinka jų lūkesčius, tačiau 27,8 % mokinių norėtų mokytis kurti programėles (apps), žaidimus ar programas praktiniams naudojimui. Šiek tiek daugiau nei dešimtadalis (11,1 %) mokinių nori papildomai mokytis antros programavimo kalbos, kiti šiuo klausimu nuomonės neturi.

Mokinių nuomonės išsiskiria ir kalbant apie privalomas programavimo pamokas I (9 kl.) ar / ir II (10 kl.) klasėse. Trečdalis mokinių tam pritartų, nes, anot jų, programavimas lavina loginį bei algoritminį mąstymą, kūrybiškumą. 22,2 % apklaustųjų tam pritaria galvodami apie IT brandos egzaminą – anksčiau pradėjus mokytis programavimo galima geriau pasiruošti jam. 16,7 % respondentų nepritartų tokiai idėjai, nes, jų nuomone, tai daugeliui neįdomus dalykas. Daugiau nei ketvirtadalis gimnazistų (27,8 %) neturi nuomonės šiuo klausimu.

Sėkmingam programavimo mokymui ir mokymuisi labai svarbi mokinių motyvacija. Motyvuoti mokinius gali įvairūs aspektai (2 pav.)



2 pav. Kas Jus labiau motyvuotų mokytis programavimo?

Anketoje mokinių buvo klausama ir apie skirtingų lyčių gebėjimą įsisavinti programavimo žinias bei įgūdžius. Daugiau nei pusė apklaustųjų (55,6 %) teigia, kad sėkmingai programuoti gali tiek merginos, tiek vaikinai. Tik vienas respondentas (5,6 %) teigia, kad programuoti vaikinams sekasi geriau nei merginoms, o kiti šiuo klausimu nuomonės nepareišė.

Apklausoje taip pat buvo siekiama sužinoti gimnazistų nuomonę apie teorinių programavimo žinių pritaikymą praktiškai. Daugiau nei pusė apklaustųjų (55,6 %) nežinojo kas yra mikrovaldikliai, likusiems programuoti šių įrenginių neteko. Tačiau apklausos respondentai mano, jog teorinių žinių pritaikymas praktiškai (50 %) motyvuotų domėtis programavimu. Daugiau nei ketvirtadalis apklausoje dalyvavusių gimnazistų (27,8 %) teigia, kad praktika tikrai paskatintų labiau domėtis programavimu. Likusiems mokiniams (22,2 %) pakanka programavimo pamokose įgytų žinių.

Mokinių buvo klausama ir apie santykį su robotika. Daugiau nei ketvirtadalis apklaustųjų (27,8 %) nežino kas tai yra. 22,2 % gimnazistų norėtų dalyvauti robotikos užsiėmimuose, o 44,4 % mokinių norėtų robotikos mokytis mokykloje.

Apibendrinus gautus tyrimo rezultatus galima teigti, kad mokiniai turi geras sąlygas mokytis programavimo tiek mokykloje, tiek namuose, o programavimo pamokose įgyjama pakankamai žinių bei suformuojami įgūdžiai. Tačiau daugelis mokinių mano, jog programavimo pamokose įgytų žinių praktinis pritaikymas didintų jų motyvaciją mokytis programavimo. Dalis respondentų pasigenda adaptuotos programavimo mokymo aplinkos, kurioje būtų galima labiau personalizuoti ar diferencijuoti ugdymo procesą parenkant tinkamą mokymosi kontekstą ar technologijas. Šių lūkesčių įgyvendinimui bus kuriama mikrovaldiklių panaudojimu grįsta programavimo mokymosi aplinka.

Skiriaus išvados

1. Sociologinio tyrimo rezultatai parodė, kad mokiniai nori įgytas teorines programavimo žinias sieti su praktika, pasigenda specializuotų programavimo mokymosi aplinkų bei įvairesnių mokymosi metodų. Tikimasi, kad kuriama mokymosi aplinka bei jos taikymas programavimui mokytis prisidės prie geresnių mokinių pasiekimų, didins motyvaciją.

2. Nors gimnazijoje III–IV klasėse programavimui mokyti skirtos 2 savaitinės pamokos, daugiau nei ketvirtadalis respondentų norėtų mokytis papildomos programavimo kalbos, kas, jų manymu, leistų pasiruošti geriau brandos egzaminui ar būsimoms studijoms.
3. Gera informacinių technologijų kabinetų techninė bazė užtikrina sklandų darbą pamokų metu, tačiau mokiniai norėtų, kad pamokose būtų mokoma įvairių įtaisų konstravimo bei jų valdymo ar robotikos užsiėmimų. Tai skatintų mokinius labiau mokytis programavimo.

3. Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos projektavimas

Programavimui mokytis bei mokytis gali būti naudojamos įvairios aplinkos. Vienos iš jų daugiau naudojamos įsivertinti, kitos programoms testuoti ir problemoms spręsti, trečios veikia per žaidimus, o ketvirtąsias galime rasti mobiliajame telefone ir juo mokytis programuoti. Tačiau jau egzistuojančioms gali būti pasiūlytos ir alternatyvios programavimo mokymo aplinkos. Viena iš tokių galėtų būti mikrovaldiklių panaudojimu grįsta programavimo mokymosi aplinka. Kai kurie mokiniai su mikrovaldiklių programavimu šiek tiek gali būti susipažinę žemesnėse klasėse, kada jiems buvo siūloma programuoti *micro:bit* kompiuteriais, suprasti technologijas bei programavimo pagrindus, taip plėtojant algoritminio mąstymo, analizės ir problemų sprendimo strategijas. Tačiau Kelmės Jono Graičiūno gimnazijoje atlikto tyrimo metu paaiškėjo, jog daugiau nei pusė (55,6 %) III–IV klasių mokinių, besimokančių programavimo, nežino kas yra mikrovaldikliai. Pastarieji kaip tik gali būti naudojami teorinėms programavimo žinioms pritaikyti praktikoje. Jau minėtame tyrime mokiniai teigė, kad teorinių žinių pritaikymas praktiškai (50 %) motyvuotų domėtis programavimu, o daugiau nei ketvirtadalis apklausoje dalyvavusių gimnazistų (27,8 %) teigė, kad praktika tikrai paskatintų labiau domėtis programavimu. Pastariesiems mokinių išsakytiems teiginiais kaip vienu iš programavimo mokymosi ar domėjimosi juo motyvatoriumi gali būti mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos naudojimas pamokų metu.

Naudojant jau minėtą mokymosi aplinką programavimui mokytis būtų naudojamas *Arduino* mikrovaldiklis, kuriuo galima valdyti daviklius (jutiklius), ekranėlius, variklius ir kitus komponentus, perduoti duomenis. *Arduino* prie kompiuterio jungiamas per USB jungtį ir programuojamas naudojant specialią kalbą, labai panašią į *C++* programavimo kalbą.

Mokiniai pagal jiems pateiktus pavyzdžius konstruotų elektros grandines bei programuotų jų veikimą. Tai būtų teorinių žinių, gautų programavimo pamokose, pritaikymas praktiškai.

Diegiant tokią programavimui mokytis skirtą aplinką būtų parengti pamokų aprašai (žr. 3 priedą), apimančys visas mokykloje dėstomas programavimo temas. Įrenginių programavimas bei valdymas tikrai motyvuotų mokinius įgyti vis naujų programavimo žinių ir pritaikyti bei įtvirtinti turimus įgūdžius rašant sudėtingesnes programas įvairiems įrenginiams valdyti. Toks praktinis pritaikymas skatintų mokinius mokytis programuoti.

3.1. Reikalavimai kuriamai mokymosi aplinkai

De Kock'as [19] teigė, kad mokymosi aplinka, kurioje naudojamas specifinis mokymosi turinys, visada padeda siekti apibrėžtų mokymosi tikslų. Planuojamoje kurti specializuotoje mokymosi aplinkoje visoms mokymosi aplinkoms būdingi komponentai dar yra papildomi specialiais mokymosi objektais (MO) ir techniniais įrankiais (mikrovaldikliais). Taip pat reikia nepamiršti, kad kuriama programavimo mokymosi aplinka. Galutiniai reikalavimai kuriamai aplinkai pateikti 3 lentelėje.

3 lentelė. Reikalavimai kuriamai mokymosi aplinkai

Reikalavimai	Komponentai	Siekiamybė
Mokymosi išteklių	<ul style="list-style-type: none"> • Įvairia forma ir formatu pateikiamas mokymosi turinys • Nuolatinis turinio atnaujinimas • Specifiniai mokymosi objektai • Įrankiai – techninė ir programinė įranga • Specialieji įrankiai – mikrovaldikliai 	<ul style="list-style-type: none"> • Aiškiai struktūriškai išdėstyti • Nuolat atnaujinami • Papildyti mokinių sukurtais
Mokymosi proceso organizavimas ir valdymas	<ul style="list-style-type: none"> • Įvairios veiklos mokymosi tikslui pasiekti • Vertinimas ir įsivertinimas 	<ul style="list-style-type: none"> • Patraukli aplinka ir lengvas naudojimas ja • Nuolatinis grįžtamasis ryšys
Bendravimas ir bendradarbiavimas	<ul style="list-style-type: none"> • Bendradarbiavimą užtikrinančios priemonės • Mokymosi objektai ir techninė įranga 	<ul style="list-style-type: none"> • Grįžtamasis ryšys
Individualus darbas	<ul style="list-style-type: none"> • Mokymosi objektai ir techninė įranga 	<ul style="list-style-type: none"> • Individualizuotas mokymasis

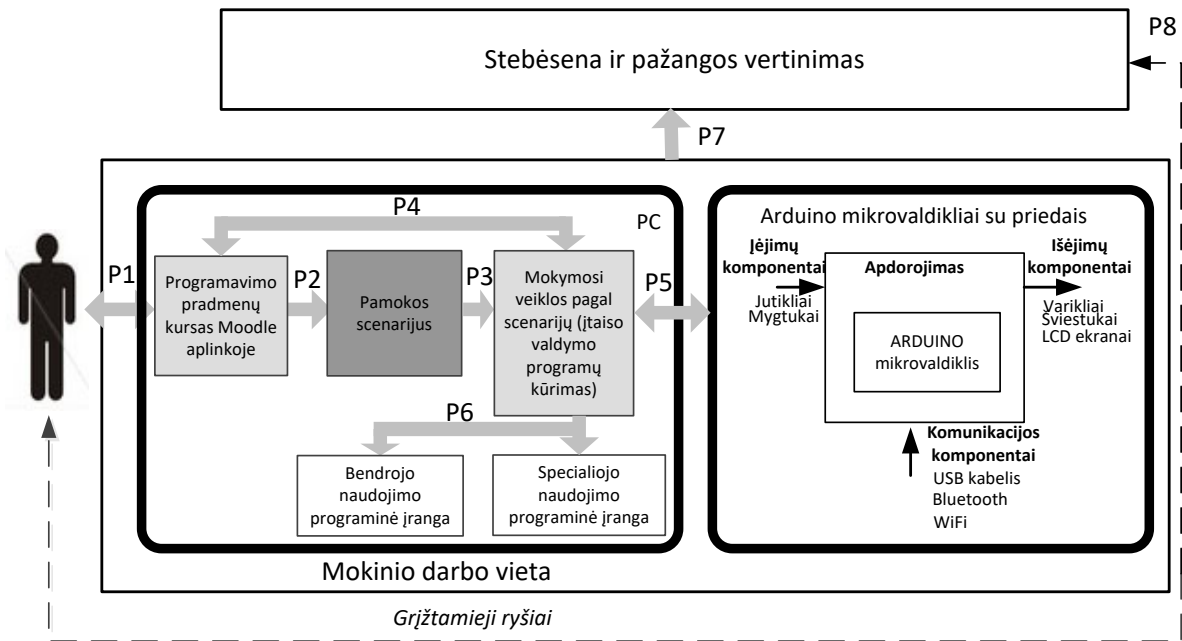
Mokymosi aplinkoje pateikiami MO turi būti glaudžiai susiję su mokymosi tikslu ir turiniu, labiausiai mokymuisi tinkamų įrankių naudojimu, mokymosi stiliaus pasirinkimu. Kiekvienas besimokantysis turi tik jam būdingas asmenines savybes bei mokymosi stilių, todėl MO turėtų sietis su besimokančiojo mokymosi stiliumi, pamokos tema bei laukiamu rezultatu.

Kalbant apie MO kūrimą, reikėtų nepamiršti ir jam keliamo pagrindinio reikalavimo – pakartotinio panaudojimo ir galimybės adaptuoti kitose mokyklose.

3.2. Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos architektūra ir funkcionalumas

Projektuojamoje mokymosi aplinkoje bus integruota VMA *Moodle*. Kuriama aplinka yra papildyta fiziniais komponentais – *Arduino* mikrovaldikliais su priedais, kas daro ją išskirtinę bei labiau patrauklią besimokančiajam. Programavimo mokymosi aplinkos dalyvio darbo vieta sudaryta iš dviejų dalių – kompiuteris su įdiegta bendrojo ir specialiojo naudojimo programine įranga ir *Arduino* mikrovaldikliai su priedais. Dalys tarpusavyje glaudžiai susijusios ir sąveikauja viena su kita panaudojant komunikacijos komponentus. Mokymosi procesų metu nuolatinis dėmesys skiriamas stebėsenai ir pažangai vertinti. Mokinys gali nuolat patikrinti savo žinias bei įgūdžius, įvertinti savo silpnąsias ir stipriąsias programavimo mokymosi sritis, tokiu būdu gauti grįžtamąjį ryšį, kurį papildo mokytojas.

3 paveiksle pateikti mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos darbo vietos komponentai nurodant jų sąveiką. Klasėje tokių darbo vietų yra tiek, kiek mokinių.



3 pav. Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos darbo vietos architektūra

3.3. Kuriamos mokymosi aplinkos procesai

Žvelgiant į projektuojamos mokymosi aplinkos architektūrą (3 pav.) matyti, kad joje bus realizuoti 8 procesai (P1-P8).

P1 proceso metu (pradinis etapas) mokinys jungiasi prie virtualioje mokymosi aplinkoje esančio „Programavimo pradmenų“ kurso, kuriame galima susipažinti (P2) su Pamokos scenarijumi.

Naudodamasis pamokos scenarijumi, mokinys atlieka (P3) mokymosi veiklas – kuria įtaiso valdymo programas. Norint jas sėkmingai atlikti gali reikėti bendrojo ar specialiojo naudojimo programinės įrangos (P6) ar papildomos informacijos, kuri pateikiama „Programavimo pradmenų“ kurse. Sąryšį tarp pastarojo kurso ir mokymosi veiklų nusako P4 procesas.

Parašytą programą mokinys įkelia į sukonstruotą įtaisą su *Arduino* mikrovaldikliu (P5). Mokinys testuoja pavyzdinę programą, daro pakeitimus programos kode.

Sėkmingam mokymosi procesui organizuoti projektuojamoje aplinkoje įdiegta Stebėseną bei pažangos vertinimas (P7). Mokiniam ypač svarbu žinoti, kaip jiems sekasi įgyti naujų žinių ar suformuoti naujus įgūdžius. Tam numatyti savikontrolės testai, kuriuos atlikdami besimokantieji stebės savo pažangą. Žinoma, pažangą mokiniai mato ir atlikdami praktines užduotis bei ieškodami problemų sprendimų būdų. Ne mažiau svarbus mokymosi procese ir Grįžtamasis ryšys (P8), kurio dėka mokiniai informuojami apie darbo rezultatus: kas pavyko geriausiai, ką reikėtų tobulinti, į ką atkreipti dėmesį.

3.4. Kurso projektavimas

4 lentelėje pateikiamas pagrindinio ugdymo bendrųjų programų pasirenkamojo programavimo mokymo(si) turinys, kuriame nurodoma, kokius įgūdžius, žinias ir supratimą turi įgyti besimokantieji. Lentelėje taip pat nurodomi uždaviniai, kurie bus siūlomi mokiniams projektuojamoje programavimo mokymosi aplinkoje.

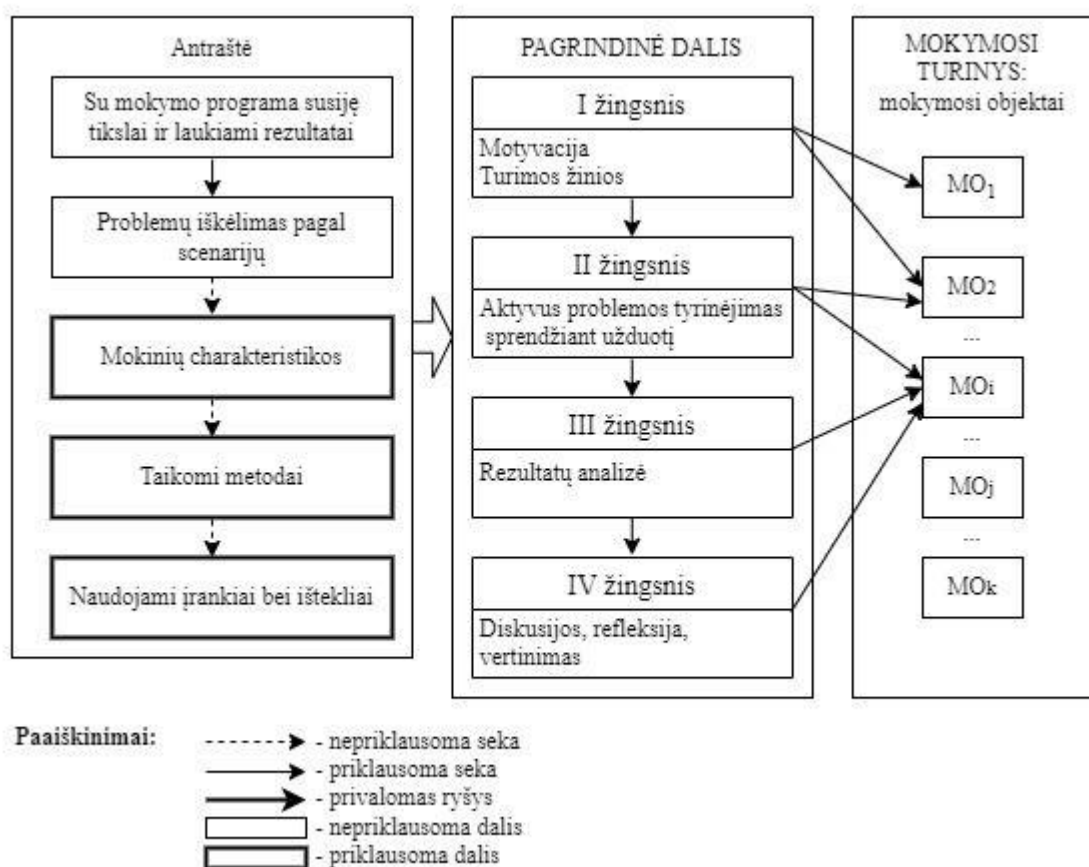
4 lentelė. Programavimo pradmenų modulyje mokinių įgyjami įgūdžiai, žinios ir supratimas

	Įgūdžiai	Žinios ir supratimas	Uždaviniai, sprendžiami naudojant mikrovaldiklius
1.	Paaishkinti algoritmo sampratą ir susieti su programavimu.	1.1. Paaishkinti, kas yra algoritmas, pateikti pavyzdzių. 1.2. Apibūdinti, kas yra programa, kaip ji susijusi su algoritmu. 1.3. Paaishkinti, kas yra programavimo kalbos, programavimo terpės, kam jos reikalingos. 1.4. Apibūdinti kompiliatoriaus paskirtį.	Darbo su <i>Arduino</i> pradžia – supažindinimas su <i>Arduino</i> mikrovaldikliu bei jo priedais, <i>Arduino</i> programavimo aplinka.
2.	Atlikti veiksmus su įvairių tipų duomenimis, skirti programos argumentus ir rezultatus.	2.1. Apibūdinti duomenų, kintamojo ir kintamojo reikšmės sąvokas. 2.2. Paaishkinti, kas yra pradiniai ir galutiniai programos duomenys. 2.3. Paaishkinti priskyrimo sakinio struktūrą, pateikti pavyzdzių. 2.4. Užrašyti veiksmus su įvairių tipų duomenimis, naudoti juos programose.	Šviestukų programavimas. Įvestis ir išvestis.
3.	Taikyti pagrindinius algoritmų veiksmus ir užrašyti juos programavimo kalbos žymenimis	3.1. Apibūdinti nuoseklų veiksmų atlikimą – veiksmų seką, pateikti pavyzdzių. 3.2. Apibūdinti veiksmų pasirinkimą – šakojimą, pateikti pavyzdzių. 3.3. Apibūdinti veiksmų kartojimą – ciklą, pateikti pavyzdzių.	Atstumo iki pasirinkto objekto radimo programavimas <i>Servo</i> varikliukų programavimas
4.	Sudaryti programas nesudėtingiems uždaviniams spręsti.	4.1. Sudaryti programas uždaviniams, taikant žinomas formules arba žinomus algoritmus, spręsti. 4.2. Parengtas programas vykdyti kompiuteriu	Triukšmo jutiklio programavimas
5.	Sprendžiant uždavinius laikytis programos sudarymo etapų.	5.1. Taikyti pagrindinį algoritmų ir programų sudarymo principą – uždavinio skaidymą į dalis. 5.2. Nusakyti pagrindinius programos parengimo etapus: rašymą, derinimą, testavimą. 5.3. Paaishkinti kontrolinių duomenų svarbą programai.	Garso signalizatoriaus <i>Pjezo</i> programavimas
6.	Laikytis programavimo kultūros principų.	6.1. Apibūdinti programavimo stiliaus ir kultūros sąvokas, pateikti pavyzdzių. 6.2. Programoje parinkti prasmingų vardų, taisyklingai juos užrašyti, vaizdžiai išdėstyti programos tekstą. 6.3. Aprašyti programoje atliekamus veiksmus komentarais.	Šviesos įjungimas ar išjungimas automatiškai reaguojant į apšvietimą

Programavimo pradmenų modulis yra papildytas realiame gyvenime naudojamų įtaisų programavimu panaudojant mikrovaldiklius. Mokiniais bus pateikiami *Arduino* programavimo aplinkoje parašyti programų kodai. Juos reikės testuoti, modifikuoti bei pagal tuos kodus parašyti visai naujas programas *Arduino* programavimo aplinkoje, kad įtaisai atliktų jiems skirtas užduotis. Kai kurios užduotys paimtos iš kitų tinklalapių <http://arduino.cc>, <https://arduinogetstarted.com/>, tačiau jos adaptuotos pagal mokinių turimas žinias ir pasiekimus.

Kalbant apie realaus pasaulio įtaisų programavimą naudojant mikrovaldiklius, reikėtų akcentuoti, kad mokiniai lavina visus pagrindinio ugdymo bendrosiose programose programavimo pradmenims įsisavinti iškeltus įgūdžius.

Apibendrintas mokymosi scenarijaus modelis pateiktas 4 paveikslėlyje.



4 pav. Apibendrintas mokymosi scenarijaus modelis [20]

Apibendrinto mokymosi scenarijaus pagrindas yra paimtas iš tyrimais pagrįsto gamtos mokslų ugdymo scenarijaus struktūros [20]. Pasirinktą specifikaciją traktuojame kaip tipinę arba bendrąją, nes jos struktūra yra įprasta daugeliui naudojimo atvejų, o straipsnyje [20] ji pateikiama kaip scenarijaus šablonas. Tačiau konstrukcijos turinys turi būti pritaikytas konkrečiai situacijai. Taigi, darome tą patį. Pirma, pradiniam scenarijuje pateikiame nedidelius struktūrinius pakeitimus. Dviejų lygių struktūrą palikome nepakeistą (4 pav. vaizduojame kaip antraštę ir pagrindinę dalį). Tačiau pagrindinėje dalyje

žingsnių skaičių sumažiname nuo penkių iki keturių. Be to, atlikome keletą pakeitimų antraštėje, t. y. neįtraukėme mokinių vaidmenų, nes jie yra vienodi, ir sujungėme laukiamus rezultatus su programavimo pradmenų mokymo programos tikslais.

Apibendrinto mokymosi scenarijaus modelio antraštėje yra penki elementai, o pagrindinėje dalyje – keturi elementai. Abiejų dalių elementų turinys yra visiškai specifinis ir orientuotas į STEM pagrįstą mokymą naudojant mikrovaldiklius. Specifiškumas atsiranda kitame mūsų sistemos procese, kai pristatome scenarijų tipus, orientuotus į STEM orientuotus tikslus ir kontekstą, išdėstytus per ugdymo ciklą – pusmetį ar net visus mokslo metus. Nurodytos sekos laikymasis nėra privalomas. Atsižvelgiant į tikslus ar kitas priežastis, kai kuriuos scenarijus galima praleisti. Pateikta seka palaiko ne tik probleminį mokymąsi, bet ir tyrimais pagrįstą mokymąsi. Šis požiūris priklauso nuo jau turimų mokinio žinių, kad galėtų pats kurti naujas. Todėl nurodyta seka skatina laipsnišką žinių, gautų naudojant ankstesnį scenarijaus tipą, naudojimą.

3.5. Skyriaus išvados

1. Sukurta mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos darbo vietos architektūra.
2. Kurso temoms specifikuoti pritaikytas apibendrintas mokymosi scenarijaus modelis.

4. Virtualios mokymosi aplinkos galimybės kuriamoje aplinkoje

Mikrovaldikliais grįstos programavimo mokymosi aplinkos darbo vietos architektūroje numatyta, kad mokinio darbo vietą sudaro komponentų aibė, tarp kurių yra ir virtuali mokymosi aplinka (VMA). Šiame skyriuje įvertinsime, kuri VMA yra tinkamiausia kuriamai aplinkai.

4.1. Virtualios mokymosi aplinkos samprata

Informacinės komunikacinės technologijos švietime yra neatsiejama mokymo bei mokymosi dalis, įgalinanti keisti ugdymo proceso organizavimą, jo dalyvių bendravimą bei bendradarbiavimą, individualizavimą, mokymosi medžiagos pateikimą, kontrolę bei savikontrolę.

Virtualioji mokymosi aplinka – tai kompiuterių tinklais ir kitomis informacinėmis komunikacinėmis technologijomis pagrįsta ugdymo sistema.

Virtualioji mokymosi aplinkos sąvoką vartojame, norėdami įvardyti vieną iš sistemų ar jų derinį:

- mokymosi valdymo sistema (anglų k. *Learning Management System*), leidžianti besimokančiųjų ar dėstytojų registravimą sistemoje ir naudojimąsi mokymosi medžiaga kompiuterių tinkle, ar sudėtinga sistema, stebinti mokinių mokymosi eigą ir pagal tai atliekanti kitas ugdymo proceso organizavimo funkcijas;
- mokymosi turinio valdymo sistema (anglų k. *Learning Content Management System*), daugiau pritaikyta individualiems poreikiams (pvz., galima stebėti, kokią mokymosi medžiagą naudojo besimokantysis, sekti pastarojo atsiskaitymus bei pagal rezultatus toliau organizuoti atitinkamą ugdymo procesą);
- turinio valdymo sistema (anglų k. *Course Management System*), leidžianti dėstytojui be HTML arba programavimo kalbos žinių parengti mokymo kursą ar kitą ugdymo procesui reikalingą informaciją;

Virtualiojoje mokymosi aplinkoje gali būti organizuojamas visas ugdymo procesas: įvairiomis formomis bei formatais pateikiamas mokymosi turinys; diskusijų forumuose, pokalbiuose ar naudojant kitas veiklas galimas dalyvių bendravimas ar bendradarbiavimas; besimokantieji gali individualiai jiems patogiu laiku bei nepriklausomai kur jie yra atlikti įvairias užduotis, savarankiškai pasitikrinti savo žinias; įgytos žinios bei gebėjimai tikrinami kompiuteriniais testais, vertinama automatinėmis priemonėmis, galima stebėti pažangą.

Pagrindiniai VMA komponentai pateikti 5 paveikslėlyje.

VMA

Mokymosi medžiagos pateikimas bei tvarkymas

Mokymosi proceso stebėjimas bei valdymas

Užduotys bei vertinimas

Bendravimas ir bendradarbiavimas

Individualus darbas

5 pav. Virtualios mokymosi aplinkos pagrindiniai komponentai

VMA leidžia mokymosi procesą padaryti kitokiu, nei įprasta tradiciniame mokyme. Naudojant VMA galima labiau individualizuoti užduotis bei jas diferencijuoti, kad besimokantieji sėkmingiau mokytųsi ir pasiektų sėkmę. Tai juos motyvuoja.

Naudojant VMA ugdymo procesas tampa lankstus ir patogus ir besimokančiajam, ir dėstytojui. Pateiktą mokymosi medžiagą galima panaudoti pakartotinai, jos pateikimas skirtingais formatais (paveikslėliai, garso ar vaizdo įrašai, failų pavidalu ir pan.) daro mokymosi procesą patrauklesniu besimokančiajam. Taip pat pažymėtina, jog mokymosi medžiagą labai lengva atnaujinti.

Virtualiojoje mokymosi aplinkoje lengva rengti įvairias užduotis, organizuoti besimokančiųjų kontrolę ar savikontrolę, stebėti pažangą, besimokančiųjų aktyvumą ugdymo procese.

Kalbant apie besimokančiųjų efektyvų mokymąsi, reikėtų nepamiršti ir kokybiško mokymosi medžiagos paruošimo ir pateikimo virtualioje mokymosi aplinkoje.

4.2. Virtualiosios mokymosi aplinkos dalyviai ir jų poreikiai

Virtualiojoje mokymosi aplinkoje organizuojant ugdymo procesą yra įvairių dalyvių, todėl jiems yra priskiriami vaidmenys su atitinkamomis teisėmis pagal poreikius.

VMA dalyviai:

- administratorius – vykdo visas leidžiamas VMA funkcijas;
- kurso kūrėjas – kuria įvairius kursus;
- dėstytojas – ugdymo procese naudoja savo sukurtą kursą, kurį gali pagal poreikius modifikuoti, valdyti (įtraukti, pašalinti, suskirstyti į grupes) kurso dalyvius ir pan.
- dėstytojas be redagavimo teisės – naudojasi jau sukurtais kursais organizuojant ugdymo procesą, atlieką kontrolę ir pan.;
- besimokantysis – gali naudotis kurso, kurio dalyvis jis yra, mokymosi medžiaga, atlikti užduotis ir pan.;

- svečias – gali naudotis kurse esančia mokymosi medžiaga.

4.3. VMA posistemiai

Kiekviena VMA – sistema, turinti pagrindines funkcijas, t. y., panaudojimo atvejai (PA), kuriuos vykdant pasiekiamas atitinkamai sistemos daliai (posistemiiui) naudotojo iškeltas tikslas. Galima išskirti tokius pagrindinius VMA posistemius:

- administravimo;
- kurso valdymo;
- mokymosi turinio pateikimo;
- mokymosi proceso organizavimo ir vertinimo;
- bendravimo ir bendradarbiavimo.

Administravimo posistemyje galima konfigūruoti sistemą, rūpintis jos saugumu, valdyti dalyvius, kursus;

Kurso valdymo posistemis įgalima pateikti kurso kūrimo užklausa, valdyti kursą, jo dalyvius;

Turinio pateikimo posistemis skirtas mokymosi medžiagos pateikimui bei modifikavimui, mokymosi medžiagos peržiūrėjimui, atsisiuntimui ar atsispausdinimui;

Mokymosi proceso organizavimo ir vertinimo posistemyje galima tvarkyti veiklas, vertinti besimokančiuosius, stebėti jų pažangą, atlikti užduotis;

Bendravimo ir bendradarbiavimo posistemyje galima rašyti, skaityti žinutes, dalyvauti diskusijų forumuose, dalyvauti vaizdo konferencijose, atlikti užduotis bendradarbiaujant.

4.4. Funkciniai ir nefunkciniai VMA reikalavimai

VMA funkcijos yra skirtos sistemos naudotojų poreikiams tenkinti. Dažniausiai vienas poreikis reiškia vieną VMA funkciją – funkcinį reikalavimą. Be pastarųjų dar būna ir nefunkciniai reikalavimai – kai kurie poreikiai susiję tik su VMA savybėmis. VMA posistemių funkciniai ir nefunkciniai reikalavimai pateikiami 5 lentelėje.

5 lentelė. Posistemių funkciniai ir nefunkciniai reikalavimai

VMA posistemis	Funkciniai reikalavimai	Nefunkciniai reikalavimai
Administravimo	<ul style="list-style-type: none"> • Įdiegti sistemą ir ją valdyti • Nustatyti sistemos parametrus • Valdyti kursus / kategorijas • Keisti sistemos kalbą • Tvarkyti papildinius (įskiepius) • Nustatyti įkeliamo failo dydį • Atnaujinti sistemą • Kurti paskyras dalyviams • Priskirti dalyviams vaidmenis • Pašalinti sistemos naudotoją • Atkurti prisijungimo slaptažodį • Įdiegti modulius • Kurti klases/ grupes • Stebėti kursų dalyvių veiklas • Formuoti ataskaitas 	<ul style="list-style-type: none"> • Paprastas sistemos valdymas • Nustatyti VMA išvaizdą
Kurso valdymo	<ul style="list-style-type: none"> • Pateikti kurso užklausa • Kurti kursą • Valdyti kurso dalyvius (pridėti, pašalinti, skirstyti į grupes) 	<ul style="list-style-type: none"> • Keisti kurso dizainą • Nustatyti temų išdėstymą • Lengva prisijungti prie sistemos • Sistemoje naudojama lietuvių kalba
Turinio pateikimo	<ul style="list-style-type: none"> • Kurti kurse temas • Pateikti mokomąją medžiagą • Redaguoti mokymosi medžiagą • Pateikti mokymosi medžiagą spausdinimui ar atsisiuntimui • Skaityti medžiagą 	<ul style="list-style-type: none"> • Parinkti geriausia formatą mokomosios medžiagos pateikimui • Lengva prisijungti prie sistemos • Sistemoje naudojama lietuvių kalba
Mokymosi proceso organizavimo ir vertinimo	<ul style="list-style-type: none"> • Kurti įvairias veiklas, užduotis • Kurti kontrolės ir savikontrolės testus • Teikti grįžtamąjį ryšį • Atlikti užduoti • Atlikti testus • Matyti gautą įvertinimą • Skaityti dėstytojo komentarą • Stebėti pažangą • Gauti grįžtamąjį ryšį 	<ul style="list-style-type: none"> • Lengva prisijungti prie sistemos • Sistemoje naudojama lietuvių kalba
Bendravimo ir bendradarbiavimo	<ul style="list-style-type: none"> • Kurti forumą • Valdyti forumo srautą • Rašyti asmeninę žinutę • Atsakyti į asmeninę žinutę • Kurti bendrus dokumentus • Nusiųsti failą • Dalyvauti tiesioginiame pokalbyje • Dalyvauti vaizdo konferencijose 	<ul style="list-style-type: none"> • Lengva prisijungti prie sistemos • Sistemoje naudojama lietuvių kalba

4.5. Virtualiosios mokymosi aplinkos panaudojimo atvejų modeliai

Kuriamoje virtualioje mokymosi sistemoje turi būti užtikrintas efektyvus mokymo bei mokymosi procesas visiems jo dalyviams.

Dalyviai gali:

Administratorius (gali būti kartu ir mokytojas) užtikrina sklandų sistemos darbą, valdo sistemoje esančius dalyvius (gali pridėti juos, pašalinti, perkelti į grupes ar klases), bendrauja su mokytojais, suteikia jiems vaidmenis.

Mokytojas kuria kursus, suskirsto besimokančiuosius į grupes jei reikia, ruošia mokymosi medžiagą, užduotis ar numato kitas veiklas, organizuoja vertinimą bei įsivertinimą, bendrauja įvairiais būdais su besimokančiais.

Besimokantysis virtualioje mokymosi sistemoje gali naudotis pateikta mokymosi medžiaga, atlikti užduotis, bendrauti su kitais dalyviais, bendradarbiauti su besimokančiais, susipažinti su savo įvertinimais ir stebėti pažangą.

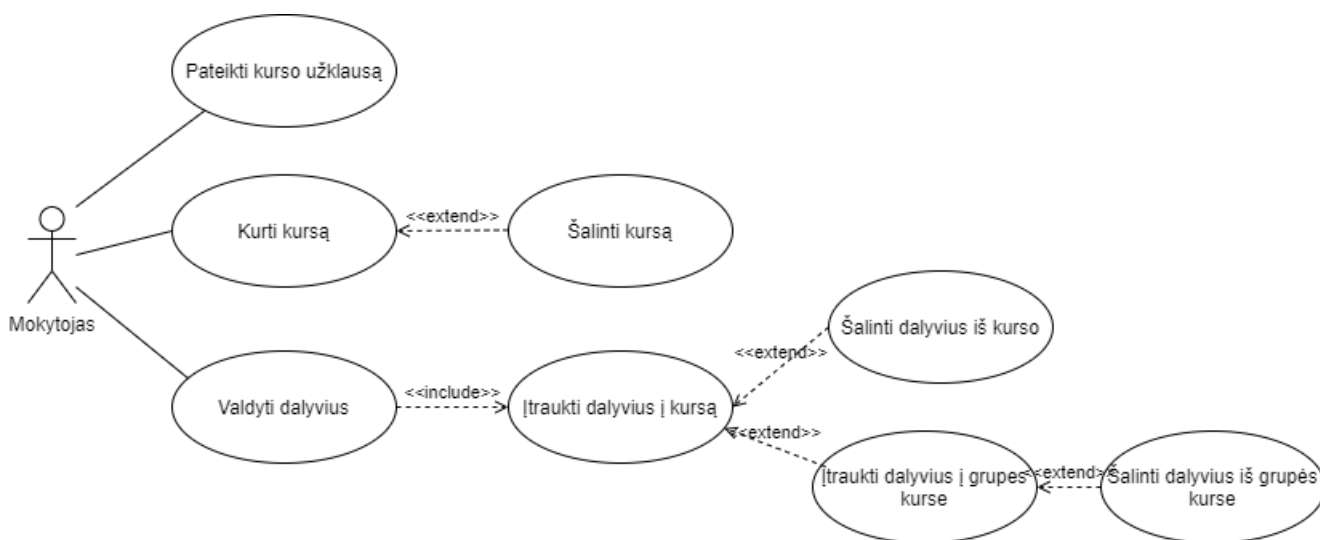
Svečias virtualioje mokymosi sistemoje gali naudotis tik pateikta mokymosi medžiaga.

Administravimo posistemiui (6 pav.) keliami funkciniai reikalavimai yra: įdiegti sistemą ir ją valdyti, nustatyti sistemos parametrus, keisti sistemos kalbą, tvarkyti papildinius, valdyti dalyvius, stebėti dalyvių veiklas, formuoti ataskaitas, šalinti kursus.



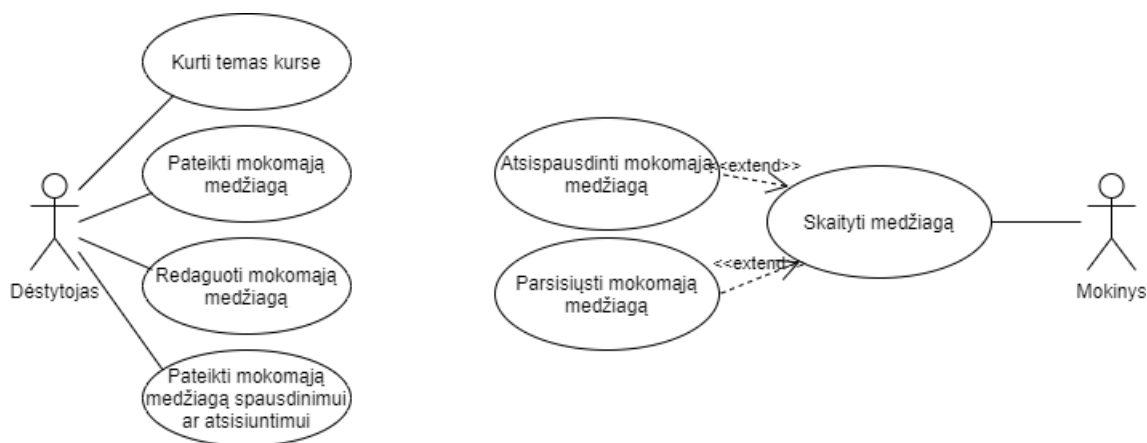
6 pav. Administratoriaus PA diagrama administravimo posistemyje

Kurso kūrimo posistemio (7 pav.) funkciniai reikalavimai yra užsakyti naują kursą, kurti kursą ir valdyti jo dalyvius.



7 pav. Mokytojo PA diagrama kurso kūrimo posistemyje

Turinio pateikimo posistemiiui (8 pav.) keliami funkciniai reikalavimai skirstomi į mokytojo – gali kurti temas kurse, pateikti ar redaguoti mokomąją medžiagą (gali būti skirta spausdinimui ar atsisiuntimui) – ir mokinio – skaityti mokomąją medžiagą, kurią gali atsispausdinti ar atsisiųsti.



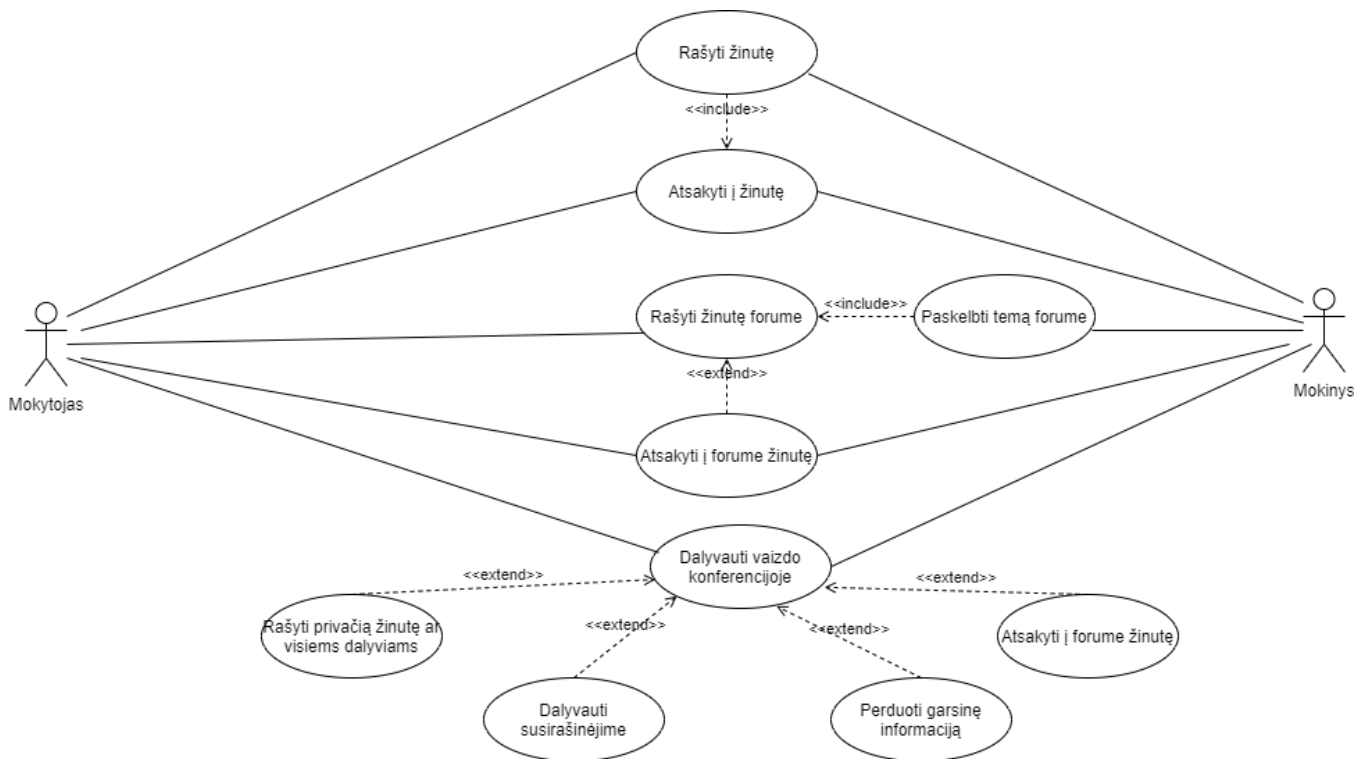
8 pav. Mokytojo ir mokinio PA diagramos turinio pateikimo posistemyje

Mokymosi proceso organizavimo ir vertinimo posistemiiui (9 pav.) keliami funkciniai reikalavimai skirstomi į mokytojo ir mokinio. Mokytojas gali kurti įvairias veiklas ar užduotis, testus ar savikontrolės testus, nevertinamas veiklas, vertinti mokinių atliktus darbus, teikti grįžtamąjį ryšį. Mokinys gali atlikti jam priskirtas užduotis, testus ar savikontrolės testus, gauti grįžtamąjį ryšį, gauti įvertinimą.



9 pav. Mokytojo ir mokinio PA diagramos mokymosi proceso organizavimo ir vertinimo posistemyje

Bendravimo ir bendradarbiavimo posistemiui (10 pav.) keliami funkciniai reikalavimai yra rašyti žinutes ar į jas atsakyti, dalyvauti diskusijų forumo veikloje, dalyvauti vaizdo konferencijose.



10 pav. Mokytojo ir mokinio PA diagramos bendravimo ir bendradarbiavimo posistemyje

4.6. Programinės įrangos parinkimas VMA realizuoti

Pagal sudarytą virtualiosios mokymosi aplinkos funkcionalumą, VMA vartotojų poreikių analizę, suprojektuotus VMA posistemius bei jų panaudojimo atvejų diagramas galima sukurti virtualiąją

mokymosi sistemą. Tai būtų ilgas procesas, reikalaujantis programavimo įgūdžių ir resursų, todėl yra tikslingiau pasinaudoti jau sukurtomis sistemomis. Egzistuoja nemažai atviro kodo ar komercinių virtualiųjų mokymosi sistemų, kuriose tai galima realizuoti.

Bandant atrinkti tinkamą mokymosi valdymo sistemą kuriamai mikrovaldiklių panaudojimu grįstai programavimo mokymosi aplinkai, buvo įdiegtos ir išbandytos pagal sudarytą projektuojamos VMA (ir jos posistemių) funkcionalumą dvi populiarios VMA: *Moodle* ir *Ilias*. Abi sistemos atitinka pagrindinius VMA keliamus funkcinis reikalavimus, jose yra realizuoti pagrindiniai įrankiai ir priemonės mokymui, mokymuisi, bendravimui ir bendradarbiavimui, tačiau abi sistemos turi ir nemažai skirtumų, kurių pagalba galima išskirti, kuri aplinka yra tinkamesnė pritaikyti kuriamai aplinkai.

6 lentelė. VMA palyginimas

VMA posistemis	Funkcinės specifikacijos	MOODLE	ILIAS
Administravimo	Sistemos nustatymų keitimas	+	+
	Valdyti kalbą	+	+/-
	Dalyvių valdymas	+	+
	Kursų kūrimas	+	+
	Ženklių valdymas	+	+
	Mobiliesiems įrenginiams	+	-
Kurso valdymo	Valdyti kursą	+	+
	Valdyti kategorijas	+	+
	Masinis kursų įkėlimas	+	-
	Valdyti dalyvius	+	+
	Masinis dalyvių įkėlimas	+	-
	Apklausų valdymas	+	+
	Klausimų banko valdymas	+	+
	Pamokų / užduočių / dokumentų valdymas	+	+
Turinio pateikimo	Atskiri tinklalapiai	+	+
	Susietų tinklalapių rinkiniai	+	+
	Minčių žemėlapis	+	+
	Apklausa	+	+
	Vaizdo ir garso įrašai	+	+
	Pateiktys	+	+
	SCORM paketų naudojimas	+	+
Mokymosi proceso organizavimo ir vertinimo	Kalendoriaus pateikimas	+	+
	Paprasta išvaizda	+	+
	Lengvai valdomas meniu	+	+
	Turinio paieška	+	+
	Įverčių knygelės valdymas	+	+
	Testas	+	+
	Testo rezultatų importavimas / eksportavimas	+	-
	Ekrano / kursoriaus padidinimas	+	-
Bendravimo ir bendradarbiavimo	Diskusijų forumai	+	+
	Pokalbis	+	+
	Žinučių rašymas	+	+
	Vikis	+	+

	Vaizdo konferencijos	+	-
	Balta lenta	+	-

Kalbant apie Administravimo posistemio funkcines specifikacijas reikėtų pastebėti, jog VMA *Ilias* nėra pilnai išversta į lietuvių kalbą, dėl ko gali būti nepatraukli sistemos naudotojui. Taip pat reikėtų nepamiršti paminėti, kad *Ilias* nėra pritaikyta mobiliems įrenginiams – tiek *Apple*, tiek *Android* naudotojams.

Kurso valdymo posistemyje daugelis galimų atlikti funkcijų abiejose sistemose sutampa, tik VMA *Ilias* negalima vienu metu įkelti kelių kursų ar kelių sistemos dalyvių.

Lyginant Turinio pateikimo posistemio funkcines specifikacijas matome, jog jos visos yra galimos abiejose VMA.

Mokymosi proceso organizavimo ir vertinimo posistemyje VMA *Ilias* negalima importuoti ar eksportuoti testo rezultatų. Mokymosi proceso organizavimas *Ilias* sistemoje kai kuriems jos dalyviams gali būti nepatrauklus ir dėl to, kad negalima keisti ekrano vaizdo dydžio.

4.7. Skyriaus išvados

1. Kuriamoje mikrovaldiklių panaudojimu grįstoje mokymosi aplinkoje įdiegta virtualioji mokymosi aplinka turi atitikti visus VMA keliamus funkcinius ir nefunkcinius reikalavimus, kad tenkintų visų sistemos dalyvių poreikius ir mokymo bei mokymosi procesą padarytų efektyviu, patraukliu.
2. Mikrovaldiklių panaudojimu grįstą programavimo mokymosi aplinką reikėtų integruoti su VMA *Moodle*. Pastaroji turi daugiau funkcinių ir nefunkcinių galimybių, kurios sudarys sąlygas labiau tenkinti kuriamos aplinkos dalyvių poreikius.
3. Dėl specifinių kuriamos mokymosi aplinkos savybių gali reikėti į VMA įdiegti papildomų modulių ar įskiepių, todėl siūloma kuriamą mokymosi aplinką susieti su VMA *Moodle*, kuri turi didelį papildomų įrankių pasirinkimą.

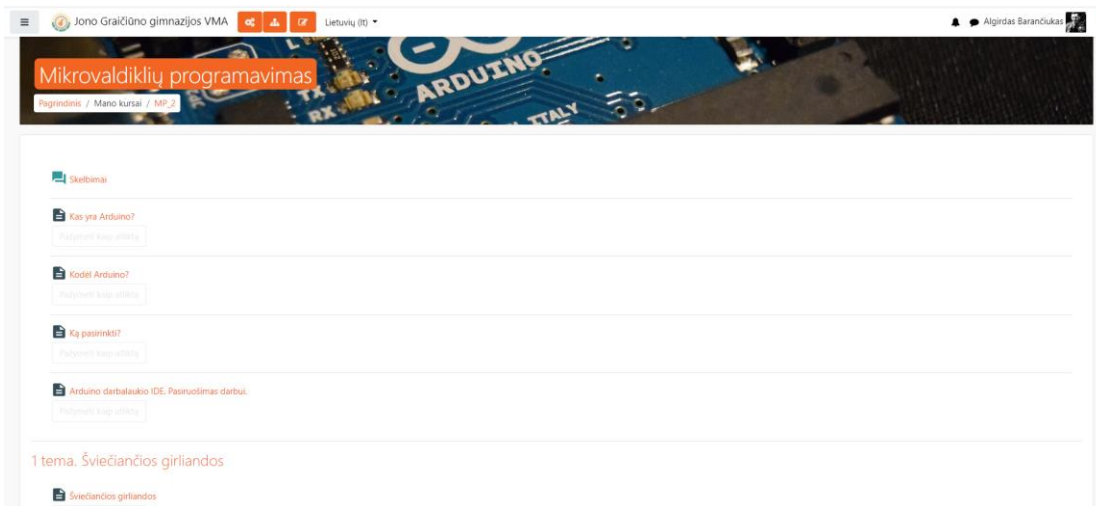
5. Kuriamos aplinkos realizavimas, vertinimas

Praėjusiame skyriuje nustatėme, kad geriausiai kuriamos mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos mokomosios medžiagos pateikimui, mokymosi proceso organizavimui, bendravimui ir bendradarbiavimui tarp mokinių ir mokytojo, grįžtamojo ryšio pateikimui geriausiai tiktų VMA Moodle. Šiame skyriuje nusakysime jos praktinį panaudojimą. Kuriamoje aplinkoje taip pat bus naudojamos ir specialiojo naudojimo programinė įranga – *Arduino* programavimo aplinka. Žinoma, bus naudojami ir mikrovaldikliai su priedais.



5.1. Kurso kūrimas Moodle aplinkoje ir jo dalyviai

Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos realizavimui pasirinkta Kelmės Jono Graičiūno gimnazija (žr. 4 priedą), o kuriamos mokymosi aplinkos programavimo pradmenų kursas sukurtas šios ugdymo įstaigos VMA Moodle, kuri pasiekiamą adresu <http://jgg.vma.lm.lt>.

Kursas Moodle aplinkoje (11 pav.) gali būti sukurtas dviem būdais. Moodle administratorius kursą sukuria tiesiogiai, pasirinkdamas atitinkamas komandas. Mokytojas pagal mokytojo PA diagramos kurso kūrimo posistemį (7 pav.) – pateikia kuriamo kurso užklausa, kurią patvirtina administratorius. Visada galima keisti sukurto VMA Moodle kurso nustatymus.



11 pav. Sukurtas kursas VMA Moodle

Kurso kūrėjas yra vienas iš kurso dalyvių, kuriam dažniausiai priskiriamas dėstytojo vaidmuo. Mokiniai į kursą registruojasi patys, jei mokytojas yra įjungęs savarankišką registraciją ( Kurso administravimas → Įregistravimo metodai → Savarankiška registracija) arba mokinius į kursą kaip dalyvius įtraukia pats mokytojas ( Kurso administravimas → Dalyviai → Įregistruoti naudotojus).

5.2. Pamokų scenarijų pateikimai

VMA Moodle kurse mokymosi medžiaga pateikiama struktūrizuotai – skirstoma temomis. Kiekvienoje temoje (12 pav.) yra pamokos scenarijus, jo versija skirta spausdinimui ar atsisiuntimui, atliktų užduočių įkėlimo galimybė. Mokymosi medžiagos pateikimas skirtingais formatais leidžia mokiniui pasirinkti tą variantą, kuris jam priimtinesnis.



12 pav. Vienos temos struktūra VMA Moodle kurse

Pamokos scenarijus VMA Moodle kurse pateikiamas naudojant išteklių *Puslapis*. Šis išteklius leidžia pateikti tekstinę (tame tarpe ir formatuotą) ar grafinę informaciją. Kiekviename pamokos scenarijuje pateikiama: pamokos tikslas, uždaviniai, naudojamos priemonės, užduotys bei mokinių veikla.

Pamokos scenarijuje pateikiami programų kodai ar kodų fragmentai, kuriuos mokiniai gali kelti į *Arduino* programavimo aplinką, įdiegtą darbo vietos kompiuteryje. Šios scenarijaus dalys (13 pav.) yra suformatuotos, išskirtos, kad mokiniams būtų lengviau atskirti, kur yra aprašymai ar užduočių tekstai, o kur – programos kodas.

2 užduotis. Kartais gyvenime reikia įjungti ar išjungti šviestuką. Papildykite programos funkciją `void loop()` keliomis eilutėmis. Nepamirškite, kad programoje naudojami kintamieji turi būti aprašyti bei inicijuoti tam skirtose programos vietose. Rezultatai stebėti valdykite **geltoną** šviestuką (10 kaištis).

```
if (analogRead(potPin) > 512){
digitalWrite(svies1, HIGH);
}
else{ digitalWrite(svies1, LOW);}
```

Paiškinimas. Įjungus šviestuką, kai vertė viršys 512 (potenciometro rankenėlės pusiaukele). Keisdami šią reikšmę, galite keisti šviestuko įjungimą ar išjungimą tik pasukus potenciometro rankenėlę ar suktelejus daugiau, o gal ir iki galo.

3 užduotis. Potenciometro pagalba galima keisti ir šviestuko ryškumą. Papildykite programos funkciją `void loop()` keliomis eilutėmis. Nepamirškite, kad programoje naudojami kintamieji turi būti aprašyti bei inicijuoti tam skirtose programos vietose. Rezultatai stebėti valdykite **raudoną** šviestuką (11 kaištis).

```
int value = analogRead(potPin) / 4;
analogWrite(svies, value);
```

13 pav. Programos kodas ar kodo fragmentai yra pateikiami suformatuoti

Kiekviename pamokos scenarijuje pateikiami sukonstruoti įtaiso elektros grandinės bei schemos paveikslėliai.

Mokiniai, kuriems nepriimtinas informacijos pateikimas kompiuterio ekrane, gali išspausdinti pamokos scenarijų – tam paruošti failai .pdf formatu.

Kiekvieno pamokos scenarijaus pabaigoje esanti paskutinė užduotis skirta savarankiškam mokinių darbui. Atlikę šią užduotį mokiniai gali įkelti į VMA Moodle nufilmuotą suprogramuoto įtaiso veikimą (naudojama veikla *Užduotis*) bei sulaukti mokytojo komentaro.

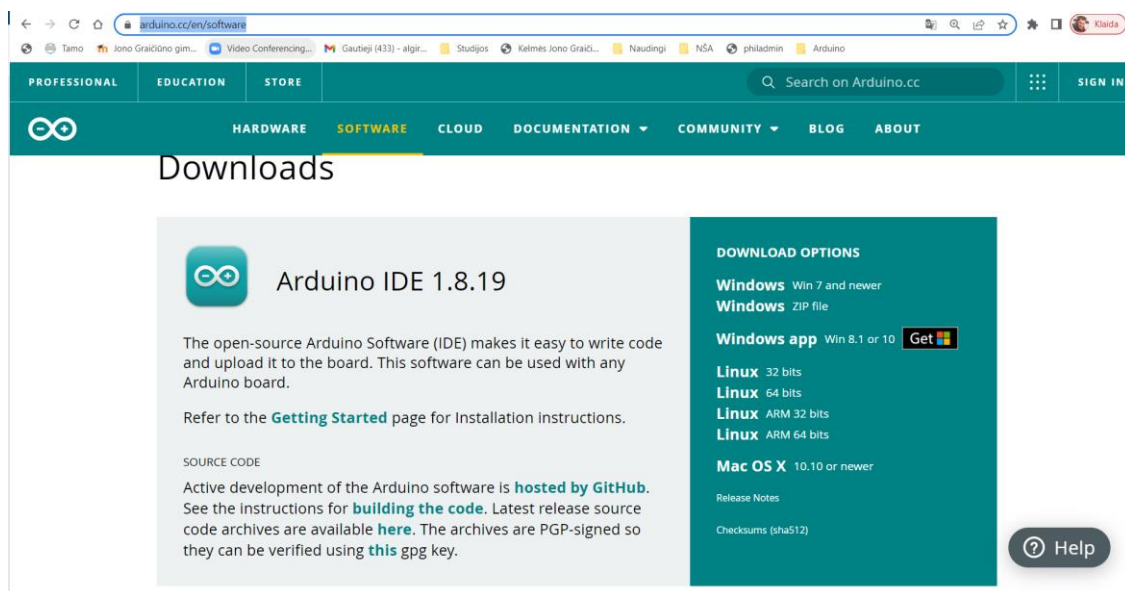
5.3. *Arduino* programinė įranga

Atliekant mokymosi veiklas pagal scenarijų reikia rašyti įtaiso valdymo programas. Tam numatyta specialiojo naudojimo programinė įranga – *Arduino* programinė įranga.

Arduino integruota kūrimo aplinka (IDE) yra programinė įranga, leidžianti rašyti kodą ir įkelti jį į *Arduino* aparatinę įrangą. *Arduino* IDE veikia *Windows*, *Mac* ar *Linux* operacinėse sistemose. Be patogių kodo redagavimo funkcijų, *Arduino* IDE yra bibliotekų, kurios suteikia papildomų funkcijų, skirtų naudoti programose, sąrašas, todėl lengviau prijungti jutiklius, ekranus, modulius ir kt.

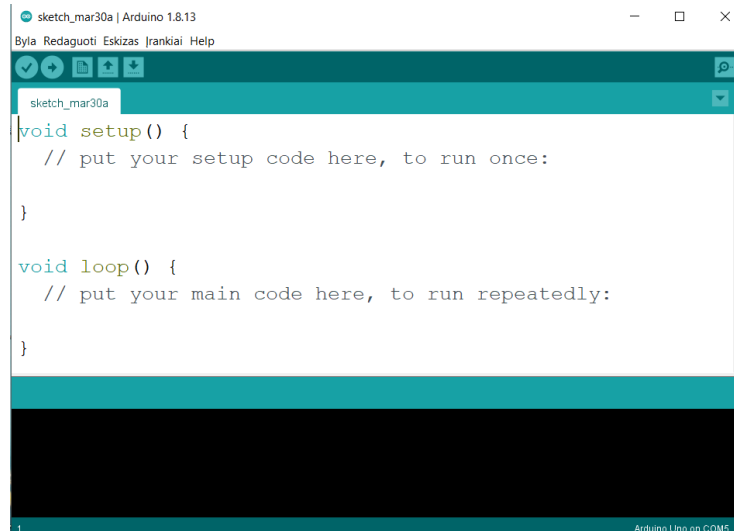
Programas galima rašyti ir įkelti jas į *Arduino* mikrovaldiklį naudojant naršyklės IDE (*Arduino Web Editor*) arba kompiuteryje įdiegtą programinę įrangą (*Arduino* IDE).

Mokiniais siūloma naudotis kompiuteryje įdiegtą *Arduino* IDE. Programos įdiegimo failas (pagal kompiuteryje naudojamą operacinę sistemą) atsisiunčiamas iš <https://www.arduino.cc/en/software> (14 pav.) ir įdiegiamas kompiuteryje.



14 pav. *Arduino* programinės įrangos įdiegimo failo pasirinkimas

Atvėrus įdiegtą *Arduino* IDE programą (15 pav.) galima rašyti įtaiso valdymo programą bei kelti ją į *Arduino* mikrovaldiklį.



15 pav. Įdiegtos *Arduino* IDE pradinis darbo langas

5.4. *Arduino* mikrovaldiklių su priedais naudojimas

Pagal kuriamos aplinkos darbo vietos architektūrą be programavimo pradmenų kurso *Moodle* aplinkoje ir bendrojo ar specialiojo naudojimo programinės įrangos dar naudojami *Arduino* mikrovaldikliai su priedais.

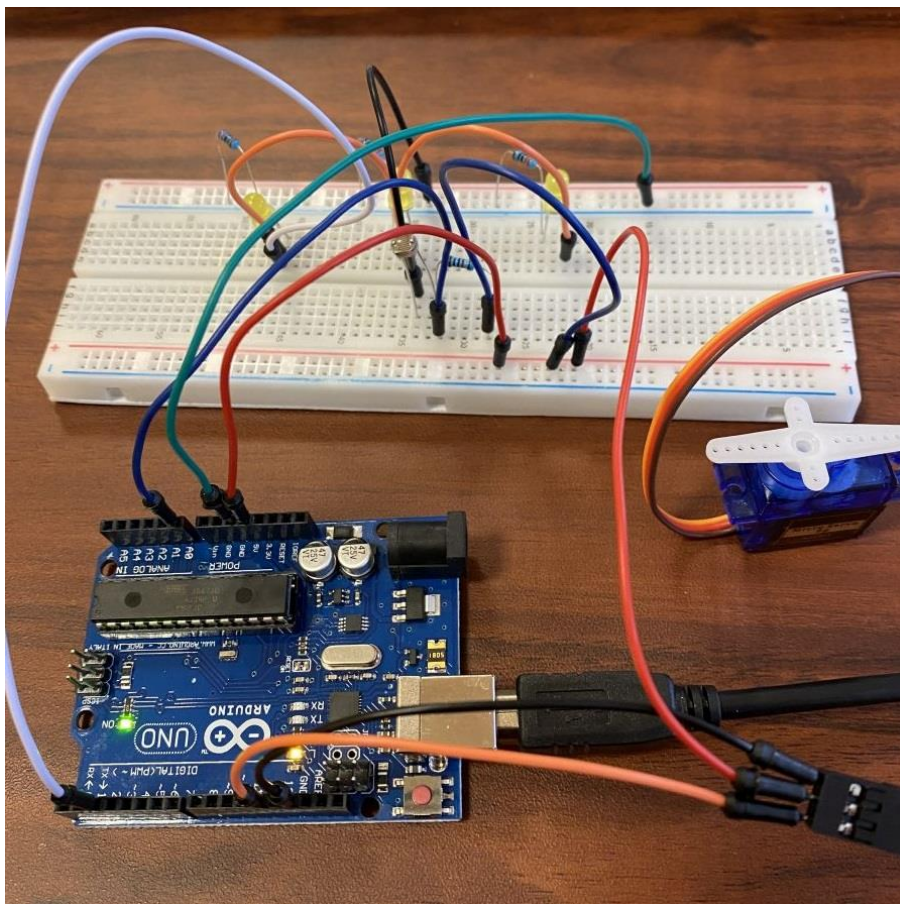
Kelmės Jono Graičiūno gimnazijoje įdiegiant mikrovaldiklių panaudojimu grįstą programavimo mokymosi aplinką nupirkti *Arduino UNO R3* rinkiniai (16 pav.). Juos naudojant galima sukonstruoti įvairius įtaisus, kuriuos paskui galima programuoti ir įgalinti veikti.



16 pav. *Arduino UNO R3* rinkinys

Pamokos scenarijuje numatytoms užduotims atlikti mokiniams pateikiami sukonstruoti įtaisai (17 pav.), kurių funkcionalumą reikia suprogramuoti – naudojant *Arduino* IDE parašyti programos kodą arba

naudoti pateiktus pavyzdžius, įkelti kodą į *Arduino* plokštę su mikrovaldikliu, kuris yra viena iš pagrindinių dalių įtaise, bei valdyti įtaisą.



17 pav. Pamokos scenarijuje numatytais užduotims atlikti pateikiami sukonstruoti įtaisai

5.5. Kuriamos mokymosi aplinkos vertinimas

Mikrovaldiklių panaudojimu grįsta programavimo mokymosi aplinka įdiegta Kelmės Jono Graičiūno gimnazijoje (žr. 4 priedą). Ją išbandė II klasės mokiniai, kurie yra pasirinkę programavimo pradmenų modulį. Norint įvertinti šios aplinkos privalumus ir trūkumus, jos naudingumą bei tobulinimo galimybes, jiems buvo pateikta anketa (žr. 5 priedą).

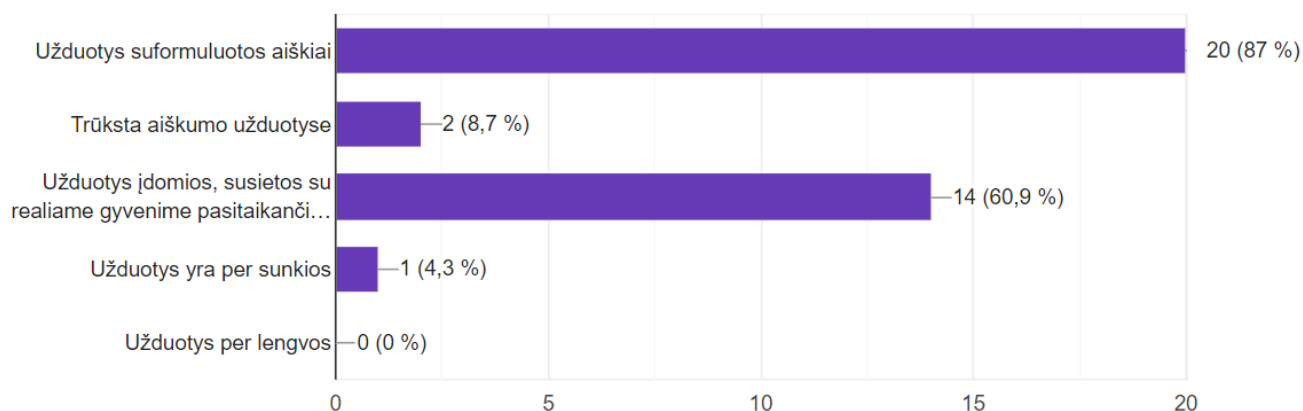
Iš viso apklausti 23 mokiniai, iš kurių 8 (34,8 %) merginos ir 15 (65,2%) vaikinių. Įdiegus mokymosi aplinką buvo sukurtas *VMA Moodle* kursas, kuriame pateikiami pamokų scenarijai ir kita informacija. Į klausimą „Ar nekyla problemų dirbant su *VMA Moodle*?“, daugiau kaip du trečdaliai respondentų (69,6 %) atsakė, kad *VMA* naudoja ne pirmus metus ir dirbti joje lengva. Dar 17,4 % apklaustųjų kaip privalumą įvardija struktūriškai bei aiškiai išdėstytą informaciją. Tačiau yra mokinių, kuriems nepatinka *VMA Moodle* (8,7 %).

Šiek tiek daugiau nei trečdalis (34,8 %) mokinių mano, kad pamokų, kuriuose mokomasi programuoti įtaisus su mikrovaldikliu, yra per mažai. Tokiam pačiam skaičiui apklaustųjų šių pamokų pakanka, o 30,8 % respondentų norėtų papildomos pamokos visus mokslo metus.

Mokinių buvo klausima apie pamokų, kuriose mokomasi programuoti mikrovaldiklius, formatą. Šiek tiek daugiau nei trečdaliui (34,8 %) apklaustųjų pamokos formatas tinkamas ir jie gali dirbti savo tempu, likusiems (65,2 %) – formatas tinkamas taip pat, o pasitaikančias problemas įveikia konsultuodamiesi su mokytoju.

Net 91,3 % respondentų labai gerai atsiliepia apie VMA Moodle aplinkoje pateikiamus pamokų scenarijus, kitą informaciją, nurodydami, jog viskas išdėstyta aiškiai ir struktūriškai, o pamokoje pateikiamų užduočių skaičius – tinkamas (100 %).

Mokiniai vertino ir pamokos scenarijuje pateiktas užduotis (18 pav.). Daugelis apklaustųjų labai palankiai vertina užduočių sąlygas, teigdami, jog jos suformuluotos aiškiai (87 %). Gerokai daugiau nei pusė respondentų (60,9 %) kaip privalumą įvardija tai, kad užduotys yra įdomios, susietos su realiaame gyvenime pasitaikančiomis situacijomis.



18 pav. Pamokos scenarijuje pateiktų užduočių vertinimas

Programavimo mokymosi aplinkoje pagal pamokos scenarijų mokiniai gauna jau sukonstruotą įtaisą – tokiam realizavimui pritaria daugiau nei pusė (52,2 %) respondentų, tačiau nemaža dalis apklaustųjų (39,1 %) norėtų patys sukonstruoti įtaisą pagal duotą schemą. Pastarajam mokinių norui įgyvendinti tektų atsisakyti dalies pamokos scenarijuje numatytų užduočių, būtų mažiau mokomasi programuoti. Dar beveik dešimtadalis (8,7 %) apklausos dalyvių nori patys išsikelti problemą, sukurti užduotis jai įveikti, sukonstruoti įtaisą, sukurti jo valdymo programą bei ištestuoti veikimą.

43,5 % apklausoje dalyvavusių programavimo pradmenų besimokančių mokinių neturėjo jokių problemų dirbdami *Arduino* IDE aplinkoje. Šiek tiek didesnė dalis (47,8 %) apklaustųjų kartais turėjo kažkokių problemų, o beveik dešimtadaliui (8,7 %) respondentų buvo sunku redaguoti pamokos scenarijuje pateiktus programų fragmentus ar rašyti naujus.

Apklausoje dalyvavo programavimo pradmenų modulį pasirinkę mokiniai, kurie mokosi C++ programavimo kalbos naudodami *Code::Blocks* aplinką. Didžioji dalis (39,1 %) respondentų manė, kad mikrovaldiklių programavime įgytos žinios ir įgūdžiai bus pritaikyti programavimo pradmenų pamokose, nes programų rašymo sintaksė ar konstruojami sakiniai panašūs abiejose aplinkose. Likę

apklausos respondentai pasidalino į dvi dalis po lygiai (po 30,4 %), teigdami, jog sunku vertinti (pirmoji grupė) ir panaudos tik nedidelį dalį įgytų žinių bei įgūdžių (antroji grupė).

Daugiau nei pusė apklausos respondentų (52,2 %) teigiamai atsakė į klausimą „Ar mikrovaldiklių programavimo siūlomoms temoms atliepia Jūsų lūkesčius?“, nes visos siūlomoms temoms jiems yra įdomios, susietos su realia pasaulyje pasitaikančiomis užduotimis. 39,1 % apklaustųjų manė, kad tik kai kurios temoms įdomios, o beveik dešimtadalis mokinių (8,7 %) pageidavo, kad temų būtų siūloma daugiau taip suteikiant pasirinkimo laisvę.

Atliekant pamokos scenarijuje pateiktas užduotis mokiniai arba patys skaito nuosekliai pamokos scenarijų ir atlieka užduotis, arba kreipiasi pagalbos į mokytoją (po 30,4 %). Šiek tiek daugiau nei trečdalis apklaustųjų (34,8 %) problemų (jei tokių būna) sprendimo ieško patys.

VMA Moodle aplinkoje be pamokų scenarijų yra pateikiama trumpa teorinė medžiaga apie mikrovaldiklių programavimą. Daugiau nei pusei apklausos dalyvių (52,2 %) pateikiamos teorinės medžiagos pakanka, o daugiau informacijos jie randa internete. Daugiau nei trečdalis respondentų (39,1 %) teorinės medžiagos norėtų daugiau, kad galėtų labiau susipažinti su mikrovaldiklių programavimu. Beveik dešimtadalis apklaustųjų (8,7 %) nori tik praktinės veiklos ir neigia teorijos svarbą.

Anketos pabaigoje apklausos dalyviai galėjo išsakyti savo pasiūlymus, pageidavimus, pastabas. Daugelis pasikartojė, jog norėtų daugiau pamokų, kuriose yra mokomasi mikrovaldiklių programavimo, nes tai jiems labai įdomu, patinka sukonstruotų įtaisų valdymas, tokia veikla juos motyvuoja.

5.6. Mokymosi aplinkos tobulinimas ir testinimas

Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos vertinimo rezultatai parodė, ką galima tobulinti ir kokios perspektyvos.

Mokymosi aplinkos funkcionalumui didinti reikia kurse pateikti daugiau informacijos apie mikrovaldiklius, jų panaudojimo galimybes konstruojant įvairius įtaisus, naudojamus realia gyvenime.

Mokymosi aplinkos VMA kurse galima pateikti įvairių tinklapių, susijusių su Arduino mikrovaldikliais, nuorodų sąrašą, kuriuo pasinaudojant besimokantieji ras papildomos informacijos, taip pat idėjų įtaisų konstravimui bei jų programavimui ar programų kodų pavyzdžių.

Tobulinant ir vystant mokymosi aplinką reikia didinti pamokų scenarijų skaičių, kad mokiniams būtų pateikiamos kuo įvairesnės užduotys.

Mokymo aplinkoje naudojami *Arduino UNO R3* rinkiniai gali būti papildomi naujais komponentais, kad būtų išplėstos konstruojamų įtaisų galimybės.

Dėl ypač palankių mokinių atsiliepimų mokymosi aplinkai, ji bus naudojama ir toliau mokantis programavimo pradmenų. Mokymosi aplinka bus pristatyta Kelmės rajono informacinių technologijų mokytojams bei pasiūlyta išbandyti ją praktiškai su savo mokiniais.

5.7. Skyriaus išvados

1. Kelmės Jono Graičiūno gimnazijoje buvo įdiegta, išbandyta ir naudojama mokymo(si) procese Mikrovaldiklių panaudojimu grįsta programavimo mokymosi aplinka.
2. Informacija apie mikrovaldiklius, sukurti pamokų scenarijai publikuojami Kelmės Jono Graičiūno gimnazijos VMA sukurtame kurse.
3. Praktinėms veikloms atlikti kompiuteriuose įdiegta *Arduino* programinė įranga bei nupirkti *Arduino UNO R3* rinkiniai.
4. Atliktas Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos vertinimas bei, pasinaudojant jo rezultatais, parengtas mokymosi aplinkos tobulinimo planas ir numatytas jos tęstinumas.

Išvados

1. Literatūros analizė parodė, kad norint besimokančiuosius skatinti domėtis programavimu reikia naudoti paprastą ir pradedantiesiems programuotojams draugišką aplinką, kuri suteiktų mokiniams savalaikį ir aktualų grįžtamąjį ryšį, būtų interaktyvi, skatintų ugdytis problemų sprendimo įgūdžius. Įveikti programavimo sunkumus dažnai padeda tinkamai parinkta programavimo kalba ir aplinka, mokymo metodai, pakankamas praktinių užduočių kiekis, kurias atlikdami mokiniai ugdomi gerus programavimo įgūdžius bei lavina loginį ir kritinį mąstymą.
2. Sociologinio tyrimo rezultatai parodė, kad mokiniai nori įgytas teorines programavimo žinias sieti su praktika, pasigenda specializuotų programavimo mokymosi aplinkų bei įvairesnių mokymosi metodų. Tikimasi, kad mikrovaldiklių panaudojimu grįsta mokymosi aplinka bei jos taikymas mokantis programavimo didins mokinių motyvaciją ir gerins programavimo žinias, ugdyt programavimo ir problemų sprendimo įgūdžius.
3. Remiantis sociologinio tyrimo rezultatais, išanalizavus virtualių mokymosi aplinkų funkcines ir nefunkcines savybes, palyginus virtualias mokymo aplinkas bei apibrėžus kuriamos aplinkos reikalavimus, sukurtas apibendrintas mokymosi scenarijaus modelis, kuris yra paremtas tyrimais pagrįsto gamtos mokslų ugdymo scenarijaus struktūra.
4. Pagal sukurtą apibendrintą mokymosi scenarijaus modelį Kelmės Jono Graičiūno gimnazijoje įdiegta, išbandyta ir naudojama mokymo(si) procese Mikrovaldiklių panaudojimu grįsta programavimo mokymosi aplinka. Ši aplinka paremta konstruktyvizmu: mokiniai dirba individualiai, žinias ir gebėjimus konstruoja laisvai, per patirtį, nebijo eksperimentuoti su iš anksto pateiktais įtaisais bei programų kodais juos modifikuojant, kas didina jų smalsumą bei motyvaciją.
5. Norint įvertinti sukurtą mokymosi aplinką, naudojamą mokymo(si) procese, besimokantiesiems buvo pateiktas klausimynas. Pagal šios apklausos rezultatus numatytas mokymosi aplinkos tobulinimas: kurse pateikti daugiau informacijos apie mikrovaldiklius, parengti daugiau pamokų scenarijų, kad mokiniai turėtų didesnę pasirinkimo laisvę, užduočių ir konstruojamų įtaisų įvairovei papildyti *Arduino UNO R3* rinkinius naujais komponentais.

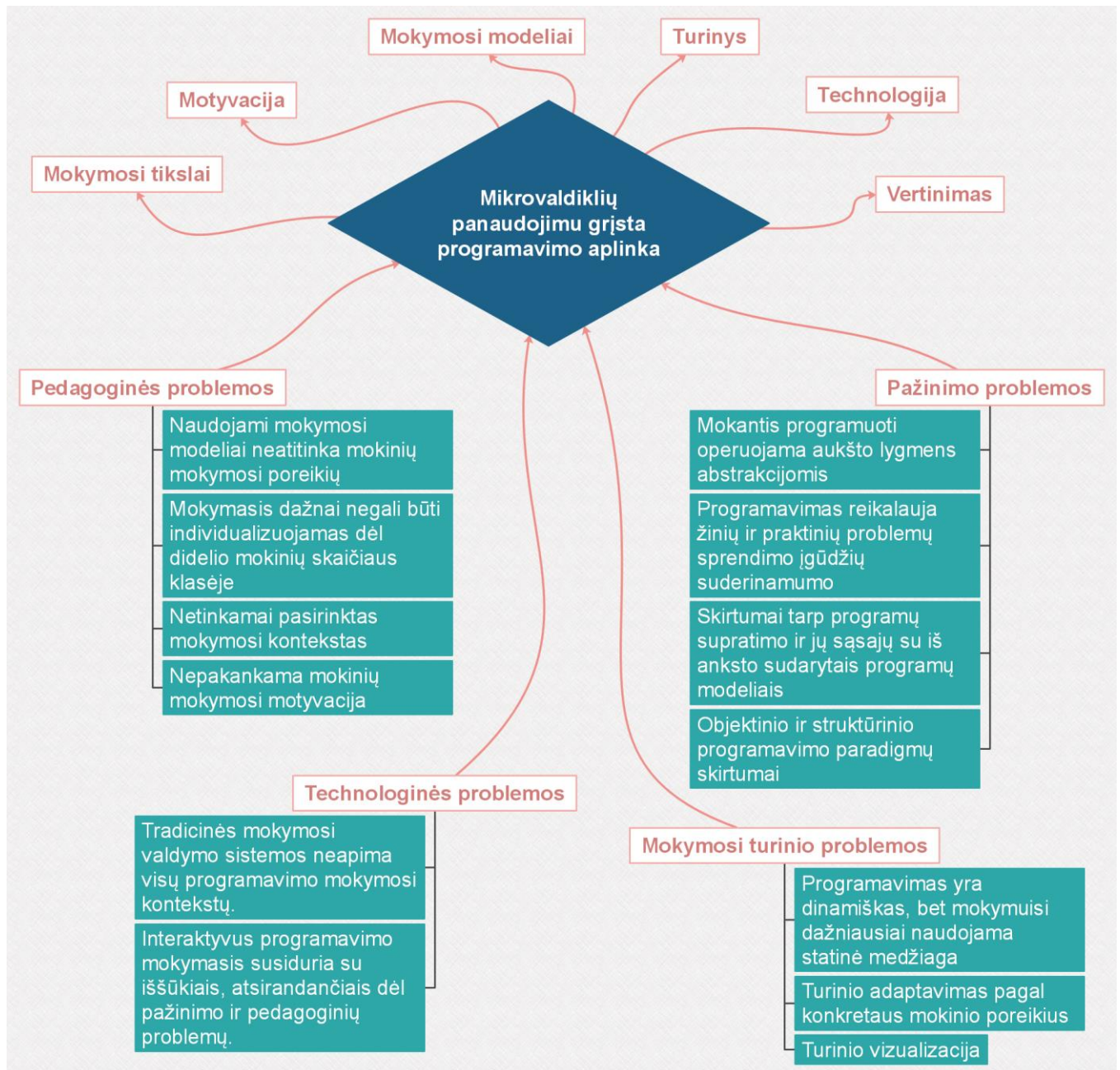
Literatūros sąrašas

1. Winslow, L. E. (1996). *Programming pedagogy—a psychological overview* [žiūrėta 2020-12-01]. Prieiga per internetą: <https://dl.acm.org/doi/pdf/10.1145/234867.234872>.
2. Dagienė, V., (2016). *Informatikos mokymo 30-metis. Ko išmokome?* [žiūrėta 2020-12-01]. Prieiga per internetą: <https://bebras.lt/wp-content/uploads/2016/08/Informatikos-mokymas-Lietuvoje-30.pdf>.
3. Gomes, A., Mendes, A. J. (2007). *An environment to improve programming education*. In Proceedings of the 2007 international conference on Computer systems and technologies (p. 88).
4. Kelleher, C., Pausch, R. (2005). *Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers*. ACM Computing Surveys (CSUR), 37(2).
5. Law, K. M. Y., Lee, V. C. S., Yu, Y. T. (2010). *Learning motivation in e-learning facilitated computer programming courses* [žiūrėta 2020-12-03]. Prieiga per internetą: <https://doi.org/10.1016/j.compedu.2010.01.007>.
6. Reardon, S., Tangney, B. *Smartphones, Studio-based learning, and scaffolding: Helping novice learn to program* [žiūrėta 2020-12-03]. Prieiga per internetą: <https://doi.org/10.1145/2677089>.
7. Sung, K., Hillyard, C., Angotti, R. L., Panitz, M. W., Goldstein, D. S., Nordlinger, J. (2011). *Game-Themed Programming Assignment Modules: A Pathway for Gradual Integration of Gaming Context Into Existing Introductory Programming Courses* [žiūrėta 2020-12-10]. Prieiga per internetą: <https://doi.org/10.1109/te.2010.2064315>.
8. Dagienė, V., Urbonienė, J. (2010). *Programavimo mokymasis: lyginamoji kalbos ir aplinkos analizė*. Informacijos mokslai, 46.
9. Mannila, L., de Raadt, M. (2006). *An Objective Comparison of Languages for Teaching Introductory Programming*. Proceedings, Koli Calling.
10. TIOBE indeksas (2022 m. balandis) [žiūrėta 2022-04-24]. Prieiga per internetą: <https://tiobe.com/tiobe-index/>.
11. *Pagrindinio ugdymo bendrosios programos: informacinės technologijos* (2008) [žiūrėta 2022-04-24]. Prieiga per internetą: https://duomenys.ugdome.lt/saugykla/bp/2016/pagrindinis/8_Informacines_technologijos.pdf.
12. *Vidurinio ugdymo bendrosios programos: informacinės technologijos* (2011) [žiūrėta 2022-04-24]. Prieiga per internetą: https://duomenys.ugdome.lt/saugykla/bp/2016/vidurinis/IT_7_priedas.pdf.

13. *Informacinių technologijų brandos egzamino programa* (2011) [žiūrėta 2022-04-24]. Prieiga per internetą: <https://www.nsa.smm.lt/wp-content/uploads/2022/01/Informaciniu-technologiju-VBE-programa-2021-10-29-V-1983.pdf>.
14. Burbaitė, R. (2014). *Išplėstiniai generatyviniai mokymosi objektai informatikos mokymuisi: koncepcija, modeliai ir realizacija*. Kaunas.
15. Djelil, F., Albouy-Kissi, A., Albouy-Kissi, B., Sanchez, E., Lavest, J. M. (2016). *Microworlds for Learning Object-Oriented Programming: Considerations from Research to Practice*. Journal of Interactive Learning Research, Association for the Advancement of Computing in Education, 27 (3).
16. Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas* [žiūrėta 2021-01-05]. Prieiga per internetą: <http://kvantti.kapsi.fi/Documents/LCL/mindstorms-chap1.pdf>.
17. Henriksen P., Kolling M. (2004). *Greenfoot: combining object visualization with interaction* [žiūrėta 2021-01-05]. Prieiga per internetą: <https://dl.acm.org/doi/pdf/10.1145/1028664.1028701>.
18. Galadima A. A. (2014). *Arduino as a learning tool* [žiūrėta 2021-01-05]. Prieiga per internetą: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6997577>.
19. De Kock, A., Slegers, P., Voeten, M. J. (2004). *New learning and the classification of learning environments in secondary education*. *Review of educational research*, 74(2).
20. Štuikys, V., & Burbaitė, R. (2018). *Smart STEM-Driven Computer Science Education: Theory, Methodology and Robot-based Practices*. Springer.

Priedai

1 priedas. Problemų medis



2 priedas. Tyrimas „Programavimo besimokančių mokinių motyvacijos didinimo galimybės taikant praktines užduotis“

Jūsų lytis?

- Mot.
- Vyr.

Kurioje klasėje mokotės?

- I
- II
- III
- IV

Kaip seniai mokotės programavimo (įskaitant mokymąsi žemesnėse klasėse, būrelių metu, savarankišką mokymąsi)

- Pirmieji metai
- Antrieji metai
- Tretieji metai
- Ketvirtieji metai
- Mokausi ilgiau nei 4 metai

Kodėl nusprendėte mokytis programavimo?

- Pasiūlė mokytojas
- Pasirinkau modulį kartu su draugais, norėdama (-as) plačiau susipažinti su programavimu
- Man patinka kompiuterių mokslas, todėl rinkausi mokytis programavimo
- Noriu būti programuotoju

Su kokiomis programavimo kalbomis teko programuoti?

- JavaScript
- Python
- C++
- C
- Perl
- C#
- PHP
- Delphi
- Ruby
- ASM
- Java

Ar programavimo mokymuisi skirtų savaitinių valandų pakanka?

- Taip, 2 savaitinių pamokų pakanka mokykliniam kursui įsisavinti III–IV klasėse
- Ne, kursas platus, todėl reikėtų tam skirti daugiau pamokų
- Neturiu nuomonės

Ar pakanka žinių, įgytų programavimo pamokų metu?

- Taip, gauname tikrai daug ir įvairios informacijos
- Ne, mokausi savarankiškai
- Ne, sandau korepetitorių

Mokydamasis programavimo naudojate?

- Pamokose gauta informacija
- Peržiūrėdamas įvairius mokomuosius vaizdo įrašus
- Nagrinėdamasis įvairius parašytus programų kodus
- Draugai, šeimos nariai padeda mokytis programavimo
- Prašau mokytojo konsultacijos

Programavimui mokytis yra palanki techninė įranga?

- Mokykloje esančių kompiuterių charakteristikos įgalina be problemų mokytis programavimo (įdiegtas kompiliatorius (*Code::Blocks*) veikia gerai)
- Namuose turiu kompiuterį, kuriame puikiai veikia programavimo mokymui įdiegtas kompiliatorius
- Namuose naudoju *online* esantį kompiliatorių – kartais būna problemų
- Turimo kompiuterio techninės charakteristikos kartais kelia problemų programavimui mokytis
- Neturiu kompiuterio namuose

Ar pakanka mokomųjų programų ar mokymui skirtų aplinkų programavimui mokytis?

- Taip, pakanka
- Mokykloje nenaudojamos jokios programavimo mokymui skirtos aplinkos
- Savarankiškai susiradau ir naudoju programavimui skirtą aplinką
- Galėtų būti daugiau adaptuotų mokyklai mokymosi aplinkų

Ar mokykloje mokytumėtės daugiau nei 1 programavimo kalbą lygiagrečiai?

- Taip
- Ne
- Neturiu nuomonės

Ar programavimo mokymo programa mokykloje atitinka Jūsų lūkesčius?

- Taip, viskas tinka
- Ne, norėčiau mokytis programuoti kitomis kalbomis
- Norėčiau mokytis kurti programėles (apps), žaidimus, programas praktiniam naudojimui
- Neturiu nuomonės

Ar reikėtų įvesti privalomas programavimo pamokas 9 ar (ir) 10 klasėse

- Taip, programavimas lavina loginį bei algoritminį mąstymą, kūrybiškumą
- Taip, nes geriau galima pasiruošti IT brandos egzaminui
- Ne, nes tai daugeliui neįdomus dalykas
- Neturiu nuomonės

Kas Jus labiau motyvuotų mokytis programavimo?

- Įdomesnės pamokos, kuriose naudojami įvairūs mokymo metodai
- Namų darbų skyrimas ir individualizavimas
- Paruoštų programų kodų su paaiškinimais pateikimas savarankiškam darbui namuose

- Programavimo mokymasis per žaidimus
- Geras mikroklimatas pamokų metu
- Draugai, besimokantys kartu
- Žinojimas, kad man reikės programavimo žinių įstojus studijuoti

Kaip programavimo žinias ir įgūdžius įsisavina skirtingų lyčių mokiniai?

- Programuoti geriau sekasi vaikinams
- Programuoti geriau sekasi merginoms
- Sėkmingai programuoti gali tiek merginos, tiek vaikinai
- Neturiu nuomonės

Kokia Jūsų patirtis programuoti mikrovaldiklius?

- Nežinau kas tai yra
- Neteko jų programuoti
- Pagrindinėje mokykloje programavau *Micro:bit*

Ar pamokose įgytų programavimo žinių pritaikymas praktiškai motyvuotų domėtis programavimu?

- Man pakanka programavimo pamokose įgytų žinių
- Būtų įdomu teorines žinias pritaikyti praktiškai
- Praktinis teorinių žinių pritaikymas tikrai paskatintų labiau mokytis programavimo

Jūsų santykis su robotika?

- Nežinau kas tai yra
- Norėčiau dalyvauti robotikos užsiėmimuose
- Dalyvauju robotikos užsiėmimuose ir man tai labai patinka
- Norėčiau robotikos mokytis mokykloje

3 priedas. Pamokos scenarijus

Vairalazdės naudojimas

PAMOKOS TIKSLAS

Išsiaiškinti vairalazdės veikimo principus ir jos panaudojimo galimybes programuojamo įtaiso valdymui.

UŽDAVINIAI

1. Susipažins kaip veikia vairalazdė;
2. Išanalizuos pateiktą programos kodą bei ištestuos, kaip tai veikia praktiškai įtaise su ARDUINO mikrovaldikliu;
3. Pakeis programos kodą ir jį pritaikys realaus gyvenimo uždaviniams spręsti;
4. Mokės įjungti/išjungti šviestukus naudojant vairalazdę.

PRIEMONĖS

Kompiuteris, įtaisas su mikrovaldikliu, šviestukais ir vairalazde, Arduino programavimo aplinka.

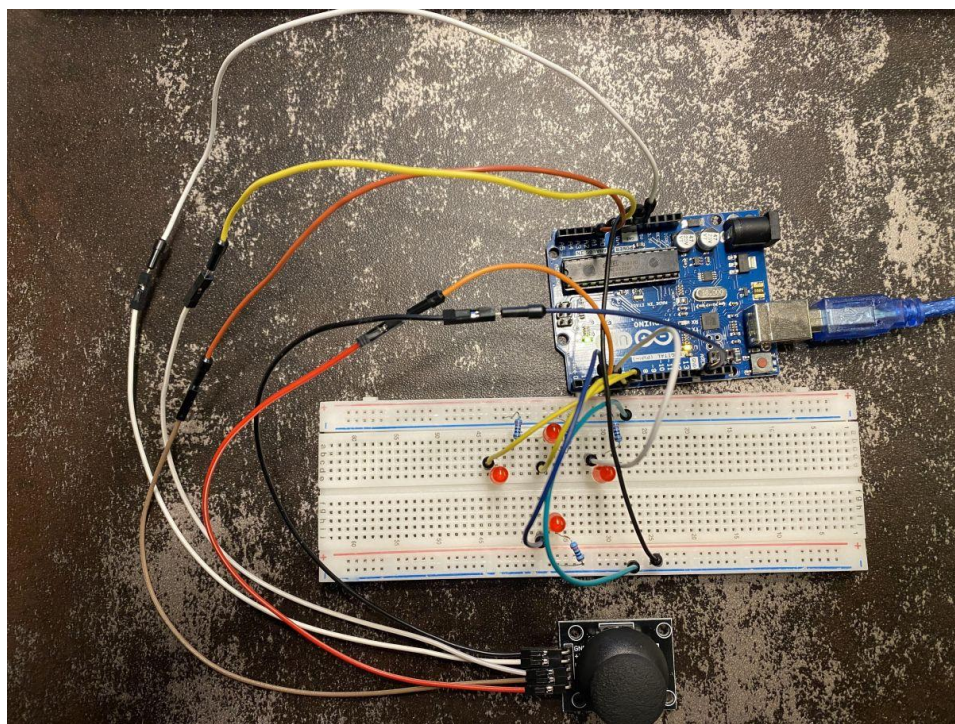
UŽDUOTYS IR MOKINIŲ VEIKLOS

Vairalazdės dažniausiai gyvenime naudojamos žaidimų ar žaislų valdymui, o kalbant apie didesnius realaus gyvenimo objektus pavyzdžiu galėtų būti ekskavatoriaus valdymas.

Prijungus vairalazdę prie mikrovaldiklio į jį yra perduodamos dvi analoginės reikšmės (nuo 0 iki 1023), atitinkančios horizontalią (x ašis) ir vertikalią (y ašis) padėtis bei viena skaitmeninė reikšmė (HIGH ar LOW) spūstelėjus vairalazdę.

Kaip veikia? Kada lenkiama vairalazdė į kairę ar dešinę, įtampa jos analoginiame VRx išvade kinta nuo 0V iki 5V (0V – kairė, 5V – dešinė). Analoginė skaitymo reikšmė proporcingai paskirstoma nuo 0 iki 1023. Kada lenkiama vairalazdė į priekį ar atgal, įtampa jos analoginiame VRy išvade kinta nuo 0V iki 5V (0V – pirmyn, 5V – atgal). Analoginė skaitymo reikšmė proporcingai paskirstoma nuo 0 iki 1023. Spūstelėjus vairalazdę perduodama skaitmeninė reikšmė, kuri gali būti HIGH arba LOW.

1 užduotis: Išnagrinėkite pateikto įtaiso (1 pav.) sudedamąsias dalis.



1 pav. Įtaisas su mikrovaldikliu, šviestukais ir vairalazde

2 uždutis

1. Atverkite Arduino programavimo aplinką ir įkelkite žemiau pateiktą programos kodą.

Programos kodas

```
/*
  Porto monitoriaus lange (Įrankiai→Porto monitorius arba Ctrl+Shift+M)
  išvedamos x ir y reikšmės
  */

#define VRX_PIN    A0 // Vairalazdės VRx išvestis prijungta prie
mikrovaldiklio analoginio A0 kaiščio
#define VRY_PIN    A1 // Vairalazdės VRY išvestis prijungta prie
mikrovaldiklio analoginio A1 kaiščio

int xValue = 0; // pradinė kintamojo xValue reikšmė
int yValue = 0; // pradinė jintamojo yValue reikšmė

void setup() {
  Serial.begin(9600) ; //Porto monitoriaus nustatymai
}
void loop() {
  // Skaitomos analoginės X ir Y reikšmės
  xValue = analogRead(VRX_PIN);
  yValue = analogRead(VRY_PIN);
  // spausdinami duomenys Porto monitoriuje Arduino IDE aplinkoje
  Serial.print("x = ");
  Serial.print(xValue);
```

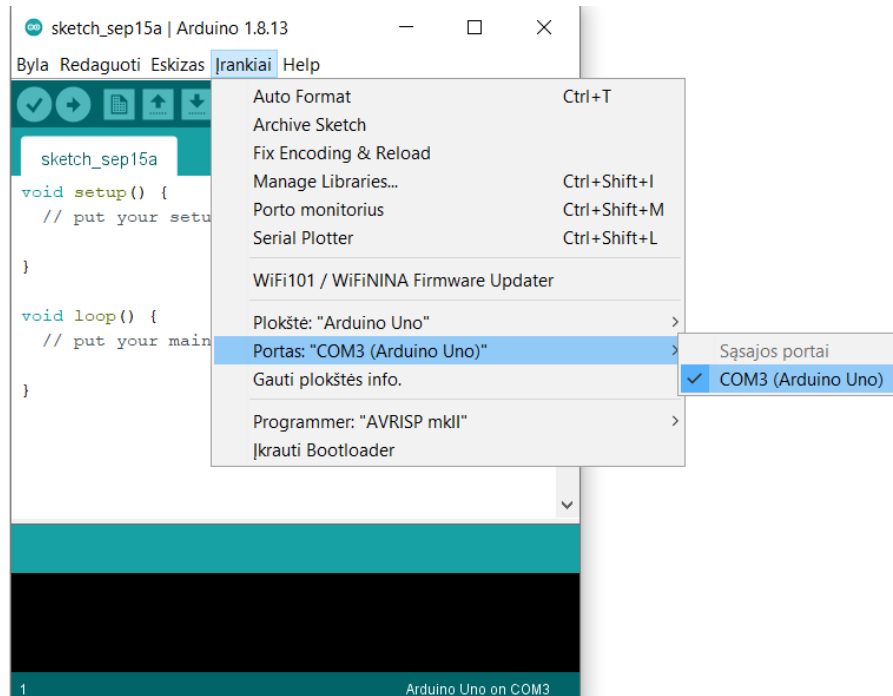


```

Serial.print(", y = ");
Serial.println(yValue);
delay(200);
}

```

2. Naudodami USB kabelį, prijunkite mikrovaldiklį prie kompiuterio ir nustatykite portą, susiejantį kompiuterį ir mikrovaldiklį (2 pav.).

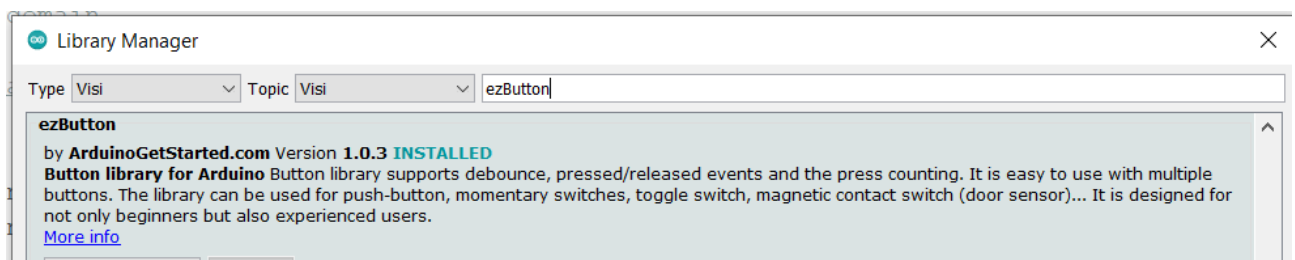


2 pav. Mikravaldiklio susiejimo su kompiuteriu porto nustatymas

3. Ištestuokite, ar pateiktas programos kodas, įkėlus jį į mikrovaldiklį, atlieka tokius veiksmus: valdant vairalazdę Porto monitoriuje yra išvedamos X ir Y reikšmės.

3 uždutis. Papildykite bei pakeiskite (jei reikia) programos kodą, kad nuspaudus ar atleidus vairalazdės mygtuką būtų atitinkami pranešimai.

1. Į kurią projektą įkelkite naują biblioteką ezButton (**Įrankiai**→**Manage Libraries...**). Atverto langelio (3 pav.) paieškos laukelyje surenkame **ezButton** ir paspaudžiame **Įdiegti**.



3 pav. ezButton bibliotekos įkėlimas į projektą

2. Nukopijuokite žemiau esantį programos kodą į Arduino IDE aplinką ir ištestuokite – stebėkite pranešimus Porto monitoriuje paspaudus vairalazdės mygtuką ar jį atleidus.

Programos kodas

```
/*
  Porto monitoriaus lange (Įrankiai→Porto monitorius arba Ctrl+Shift+M)
  išvedamos x ir y reikšmės
  */
#include <ezButton.h>

#define VRX_PIN A0 // Vairalazdės VRx išvestis prijungta prie mikrovaldiklio
analoginio A0 kaiščio
#define VRY_PIN A1 // Vairalazdės VRY išvestis prijungta prie mikrovaldiklio
analoginio A1 kaiščio
#define SW_PIN 2 // Vairalazdės SW išvestis prijungta prie 2-ojo
skaitmeninio kaiščio

ezButton button(SW_PIN);

int xValue = 0; // pradinė kintamojo xValue reikšmė
int yValue = 0; // pradinė kintamojo yValue reikšmė
int bValue = 0; // pradinė mygtuko reikšmė

void setup() {
  Serial.begin(9600) ; //Porto monitoriaus nustatymai
  button.setDebounceTime(50); // set debounce time to 50 milliseconds
}
void loop() {
  button.loop(); // Pirmiausiai PRIVALOMA įvykdyti loop() funkcija

  // Skaitomos analoginės X ir Y reikšmės
  xValue = analogRead(VRX_PIN);
  yValue = analogRead(VRY_PIN);

  // Skaitoma mygtuko reikšmė
  bValue = button.getState();

  if (button.isPressed()) {
    Serial.println("Mygtukas nuspaustas");
  }

  if (button.isReleased()) {
    Serial.println("Mygtukas atleistas");
  }

  // spausdinami duomenys Porto monitoriuje Arduino IDE aplinkoje
  Serial.print("x = ");
  Serial.print(xValue);
  Serial.print(", y = ");
  Serial.print(yValue);
  Serial.print(" : vairalazdės mygtukas = ");
  Serial.println(bValue);
}
```

4 užduotis. Papildykite programos kodą (arba nukopijuokite žemiau esantį), kurio paskirtis – analogines reikšmes pakeisti komandomis KAIRĖN / DEŠINĖN / PIRMYN / ATGAL. Ištestuokite įtaiso veikimą.

Programos kodas

```
/*
  Porto monitoriaus lange (Įrankiai→Porto monitorius arba Ctrl+Shift+M)
išvedamos x ir y reikšmės
*/

#define VRX_PIN A0 // Vairalazdės VRx išvestis prijungta prie mikrovaldiklio
analoginio A0 kaiščio
#define VRY_PIN A1 // Vairalazdės VRY išvestis prijungta prie mikrovaldiklio
analoginio A1 kaiščio

#define LEFT_THRESHOLD 400
#define RIGHT_THRESHOLD 800
#define UP_THRESHOLD 400
#define DOWN_THRESHOLD 800

#define COMMAND_NO 0x00
#define COMMAND_LEFT 0x01
#define COMMAND_RIGHT 0x02
#define COMMAND_UP 0x04
#define COMMAND_DOWN 0x08

int xValue = 0; // pradinė kintamojo xValue reikšmė
int yValue = 0; // pradinė kintamojo yValue reikšmė
int command = COMMAND_NO;

void setup() {
  Serial.begin(9600) ; //Porto monitoriaus nustatymai
}
void loop() {

  // Skaitomos analoginės X ir Y reikšmės
  xValue = analogRead(VRX_PIN);
  yValue = analogRead(VRY_PIN);

  // konvertuoti analogines reikšmes į komandas
  command = COMMAND_NO; // pradinė komandos būseną

  // komandų KAIRĖN/DEŠINĖN tikrinimas
  if (xValue < LEFT_THRESHOLD)
    command = command | COMMAND_LEFT;
  else if (xValue > RIGHT_THRESHOLD)
    command = command | COMMAND_RIGHT;

  // komandų PIRMYN / ATGAL tikrinimas
  if (yValue < UP_THRESHOLD)
    command = command | COMMAND_UP;
  else if (yValue > DOWN_THRESHOLD)
    command = command | COMMAND_DOWN;

  // Pastaba: Vienu metu gali nebūti nei vienos komandos arba būti viena ar
dvi komandos

  // Komandų išvedimas į Porto monitorių ar papildomų procesų vykdymas
  if (command & COMMAND_LEFT) {
    Serial.println("KAIRĖN");
  }
}
```

```
    // čia galima rašyti papildomus veiksmus vairalazdės padėčiai nusakyti
}

if (command & COMMAND_RIGHT) {
    Serial.println("DEŠINĖN");
    // čia galima rašyti papildomus veiksmus vairalazdės padėčiai nusakyti
}

if (command & COMMAND_UP) {
    Serial.println("PIRMYN");
    // čia galima rašyti papildomus veiksmus vairalazdės padėčiai nusakyti
}

if (command & COMMAND_DOWN) {
    Serial.println("ATGAL");
    // čia galima rašyti papildomus veiksmus vairalazdės padėčiai nusakyti
}
}
```

5 uždutis. Papildykite ar pakeiskite (jei reikia) programos kodą, kad judinant vairalazdę užsidegtų šviestukai (vairalazdės svirtis pasukta kairėn – šviečia kairysis šviestukas, dešinėn – dešinysis, į priekį – viršutinis, atgal – apatinis). Trumpai nufilmuokite šviestukų švietimą reaguojant į vairalazdės svirties padėtį ir įkelkite į žemiau esančią užduotį.

Apibendrinimas. Tai tik keletas pavyzdžių, parodančių vairalazdės galimybes bei jos panaudojimo atvejus realiame gyvenime. Panaudojimo atvejų gali būti daug daugiau. Kokie jie? Pamažtykite.

**4 priedas. Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos diegimo aktas
Kelmės Jono Graičiūno gimnazijoje**



KELMĖS JONO GRAIČIŪNO GIMNAZIJA

Biudžetinė įstaiga. Raseinių g. 1, 86162 Kelmė, tel. (8 4) 276 1195, faks. (8 4) 276 11 95

el. p. info@jggimnazija.lt

Duomenys kaupiami ir saugomi Juridinių asmenų registre, kodas 190091584

Kauno technologijos universitetui

2022-04-12 Nr. (1.23.) S-30

PAŽYMA DĖL MOKYTOJO ALGIRDO BARANČIUKO VEIKLOS

Kelmės Jono Graičiūno gimnazija pažymi, kad Kelmės Jono Graičiūno gimnazijos informacinių technologijų mokytojas metodininkas Algirdas Barančiukas gimnazijoje įdiegė, išbandė ir naudoja ugdymo procese Mikrovaldiklių panaudojimu grįstą programavimo mokymosi aplinką.

Gimnazijos direktorius



Rimas Bielskis

5 priedas. Mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos naudojimo tyrimas

Esu Kauno technologijos universiteto Nuotolinio mokymosi informacinės technologijos studijų programos magistrantas. Atlieku tyrimą, kurio tikslas – įvertinti mikrovaldiklių panaudojimu grįstos programavimo mokymosi aplinkos privalumus ir trūkumus, jos naudingumą bei tobulinimo galimybes. Maloniai prašyčiau atsakyti į žemiau pateiktus klausimus. Anketa yra anoniminė. Gauti tyrimo rezultatai bus naudojami moksliniams tikslams, rengiant baigiamąjį projektą. Nuoširdžiai dėkoju už sutikimą atsakyti į visus anketos klausimus bei už sugaištą laiką. Sėkmės!

1. Jūsų lytis
 - a. Vyr.
 - b. Mot.
2. Ar nekyla problemų dirbant su VMA Moodle?
 - a. Ne, viskas aišku, nes naudojuosi ja ne pirmus metus
 - b. Sukurtame kurse viskas išdėstyta struktūriškai, aiškiai
 - c. Kartais pamirštu prisijungimo duomenis
 - d. Nepatinka VMA Moodle
3. Ar užtenka pamokų, kuriose mokomasi programuoti įtaisus su mikrovaldikliais?
 - a. per mažai
 - b. pakanka
 - c. per daug
 - d. galėtų būti papildoma pamoka mikrovaldikliams programuoti
4. Ar tinkamas mikrovaldiklių programavimui naudojamas pamokų formatas?
 - a. Taip, galima dirbti pagal pamokos scenarijų savo tempu
 - b. Taip, iškilus problemoms visada galima kreiptis į mokytoją
 - c. Ne, man labiau patinka, kai mokytojas pamokoje pasako ką daryti
5. Ar aiški pateikiamo pamokos scenarijaus struktūra?
 - a. Taip, viskas išdėstyta aiškiai, struktūriškai
 - b. Trūksta aiškumo pamokos scenarijuje
6. Ar pateikiamas tinkamas užduočių skaičius?
 - a. Pateikiama per daug užduočių
 - b. Užduočių skaičius atitinka pamokos trukmę
 - c. Pateikiama per mažai užduočių
7. Kaip vertinate pamokos scenarijuje pateikiamas užduotis?
 - a. Užduotys suformuluotos aiškiai
 - b. Trūksta aiškumo užduotyse
 - c. Užduotys įdomios, susietos su realiame gyvenime pasitaikančiomis situacijomis
 - d. Užduotys yra per sunkios
 - e. Užduotys per lengvos
8. Kaip vertinate tai, kad gaunate užduotims atlikti sukonstruotą įtaisą?
 - a. Mane tai tenkina, nes pamokoje būtų mažai laiko sukonstruoti įtaisą ir atlikti visas pamokos scenarijuje numatytas užduotis
 - b. Norėčiau pats susikonstruoti įtaisą pagal duotą schemą

- c. Norėčiau pats išsikelti problemą, sukurti užduotis jai įveikti, sukonstruoti įtaisą, jį užprogramuoti ir ištestuoti jo veikimą
9. Ar turėjote problemų su *Arduino IDE*?
- a. Ne, viskas aišku
 - b. Kartais
 - c. Sunku rašyti programų kodus mikrovaldikliams programuoti, nes daug naujų funkcijų
10. Ar žinias ir įgūdžius, įgytus per mikrovaldiklių programavimą, pritaikysite programavimo pradmenų pamokose?
- a. Taip, nes panaši programų rašymo sintaksė, naudojamos konstrukcijos
 - b. Taip, tačiau nedidelę dalį
 - c. Ne, man tai programavimo pradmenų pamokoms jokios pridėtinės vertės neduos
 - d. Sunku vertinti
11. Ar mikrovaldiklių programavimo siūlomos temos atliepia Jūsų lūkesčius?
- a. Taip, visos siūlomos temos įdomios, susietos su realiame pasaulyje pasitaikančiomis užduotimis
 - b. Kaip kurios temos įdomios
 - c. Norėčiau visai kitokių projektinių veiklų
 - d. Temų galėtų būti siūloma daugiau, o mes galėtume iš jų rinktis, kurias užduotis atlikti
12. Ar dažnai reikia mokytojo pagalbos atliekant pamokos scenarijuje pateiktas užduotis?
- a. Stengiuosi viską atlikti pats naudojantis pamokos scenarijumi
 - b. Ieškau pats problemų sprendimo, jei tokių atsiranda
 - c. Visada kreipiuosi į mokytoją pagalbos
 - d. Problemas padeda išspręsti klasiškai
13. Ar VMA *Moodle* aplinkoje reikėtų pateikti daugiau teorinės medžiagos apie mikrovaldiklių programavimą?
- a. Taip, norėčiau plačiau susipažinti su *Arduino*
 - b. Informacijos pakanka, daugiau jos galima rasti internete
 - c. Man įdomu tik praktinė veikla
14. Jūsų pasiūlymai, pageidavimai, susiję su mikrovaldiklių programavimu.