



**VILNIAUS VERSLO KOLEGIJA**

**INFORMACINIŲ TECHNOLOGIJŲ KATEDRA  
PROGRAMAVIMO IR INTERNETINIŲ TECHNOLOGIJŲ STUDIJŲ PROGRAMA**

**AUGALO AUGINIMO AUTOMATIZAVIMAS NAUDOJANT  
ARDUINO PLATFORMĄ**

**MOKOMOSIOS PRAKTIKOS ATASKAITA**

Darbą atliko: **Povilas Svetobas**  
Darbo vadovas: lekt. dr. **Aivaras Čechamiras**

Darbas originalus \_\_\_\_\_ Vardenis Pavardenis  
(*studento parašas*)

Vilnius, 2018

## **TURINYS**

ĮVADAS	3
1. ĮRENGINIO FUNKCINIS APRAŠYMAS	4
2. TECHNINĖ ĮRENGINIO REALIZACIJA	4
2.1. Arduino komponentų aprašymas	4
2.2. Arduino komponentų sujungimo schema	6
2.3. Konstrukcinės dalies komponentų panaudojimas	7
2.4. Konstrukcinės dalies schema	8
3. PROGRAMINĖ DALIS	8
3.1. Įgyvendintų klasių ir metodų aprašymas	8
4. ĮRENGINIO TESTAVIMAS, DEFEKTŲ ŠALINIMAS	9
5. ĮRENGINIO PLĖTOTĖ	10
IŠVADOS	11
LITERATŪROS SĄRAŠAS	12

## IVADAS

**Darbo aktualumas.** Šiuolaikinio gyvenimo tempas žmogui sukelia daug iššūkių. Vienas iš garsiausių šių laikų sociologų Thomas Hyllandas Eriksenas teigia, kad lėtas laikas virsta vis didesne prabanga šiame informacijos ir begalinio vartojimo amžiuje. Para, kaip ir prieš šimtus metų sudaro dvidešimt keturias valandas, tačiau žmogus, suskaldo ją į daugybę intervalų ir užkemša parą įvairiausia veikla (Eriksen, 2004).

Taigi, vis dažniau susiduriama su situacija, kai žmogus yra nuolat užimtas ir neužtenka laiko pasirūpinti savo buitimi. Tačiau, šio laikotarpio pagrindinis bruožas yra informacinė, žinių visuomenė. Technologijos tapo mūsų neatsiejama kultūros dalis: sukurtos ir kuriamos technologijos yra naudojamos visose gyvenimo srityse, o tinkamai panaudotos technologijos gali padėti taupyti laiką - automatizuoti buitinius procesus, kurti gražią žmogaus gyvenimo aplinką.

Augalai - yra perliukai, darantys žmogaus gyvenimo aplinką gyvesnę ir jaukesnę, o Arduino mikrovaldikliai gali augalų auginimą ne tik automatizuoti, bet ir padaryti jį daug tikslesnį. Nes nesiremiama subjektyviais žmogiškais-jusliniais metodais, o augalo auginimo procesas remiasi gautais duomenimis, kuriuos interpretuoja augalui auginti skirtas algoritmas.

Šis projektas yra skirtas automatiniam eglutės auginimo įrenginiui sukurti. Aurakarija (auginamos eglutės veislė) yra inoringas ir lepus augalas. Ši eglutė nemėgsta daug saulės spindulių, laistyti aurakarija reikia saikingai, o tai reiškia, kad žemė vazone turi būti drėgna, bet ne šlapia. Augalą svarbu apipurkšti bent keletą kartų per dieną (Jakubauskaitė, 2017). Tokias sąlygas užtikrinti sunku biuro patalpose, ypač kai ne visada yra galimybė laiku prižiūrėti eglutę.

**Darbo tikslas** - Automatizuoti augalo auginimą.

### **Darbo uždaviniai:**

1. Susipažinti su Arduino platformos esamomis galimybėmis ir potencialu automatizuojant buitinius procesus;
2. Sukonstruoti automatinį augalo auginimo įrenginį;
3. Atlikti įrenginio testavimą.

### **Vartojamos sąvokos:**

1. Arduino - tai skaičiavimo platforma paremta mikrovaldiklio plokštė, turinti eilę analoginių bei diskretinių įėjimų ir išėjimų ir galinti atlikti įprasto valdiklio funkcijas. Arduino mikrovaldiklio plokštė programuojama C ir C++ pagrindais paremta programavimo kalba ir yra atviro kodo (open- source).
2. Mikrovaldiklis - integralinė schema, skirta valdyti elektros įrenginius.
3. Asembleris - programa, transliuojanti asemblerio kalba rašytą programinį kodą į mašinos kodą.
4. LED (Light - emitting diode) - šviesos diodas, kurio sandara pritaikyta šviesai skleisti.

# 1. ĮRENGINIO FUNKCINIS APRAŠYMAS

Kuriant įrenginį, buvo keliami šie funkciniai reikalavimai:

1. Įrenginys automatizuoja augalo priežiūrą.
  - Mikrovaldiklis, gavęs duomenis juos interpretuoja ir vykdo vandens padavimą į:
    - a) šakų drėkinimo sistemą (šakų apipurškimas);
    - b) šaknų drėkinimo sistemą.
  - Augalo šakų drėkinimo sistema priklauso nuo aplinkos temperatūros. Funkcionalumas, atjungiantis augalo apipurškimą, jeigu aplinkos temperatūra yra žemesnė už deklaruotą programoje reikšmę;
  - Įrenginys užtikrina, kad augalas gautų pakankamą kiekį UV spindulių, naudojant pilno spektro UV LED apšvietimą, skirtą agams auginti.
2. Įrenginys nereikalauja daug priežiūros.
  - Įrenginio priežiūrėtojas, tiesiogiai nedalyvauja augalo auginimo procese. Žmogus tik užtrina, kad:
    - a) Būtų pakankamai vandens talpykloje;
    - b) Pasirūpina elektros matinimu.
3. Konstrukcinė įrenginio dalis gali būti adaptuojama prie naujų augalo augimo sąlygų: kai augalas tampa aukštesnis, konstrukcija yra pakeliama pagal augalo aukštį. Tai užtikrina kad augalo apipurškimas bei UV LED apšvietimas vykdytų tinkamai savo funkcijas.

## 2. TECHNINĖ ĮRENGINIO REALIZACIJA

Projekto techninę realizaciją sudaro konstrukcinis - mechaninis įrenginys bei Arduino mikrovaldiklis su periferinę įranga bei kiti Arduino komponentai. Šiame skyriuje yra aprašoma įrenginio techninė realizacija.

### 2.1. Arduino komponentų aprašymas

Projekte naudojama Arduino UNO R3 mikrovaldiklio polokštė, turinti įinstaliuotą 8-bitų ATmega 328 mikrovaldikį. Taipogi, naudojami 1 lentelėje nurodyti komponentai.

**1 lentelė.** Projekte naudojami komponentai

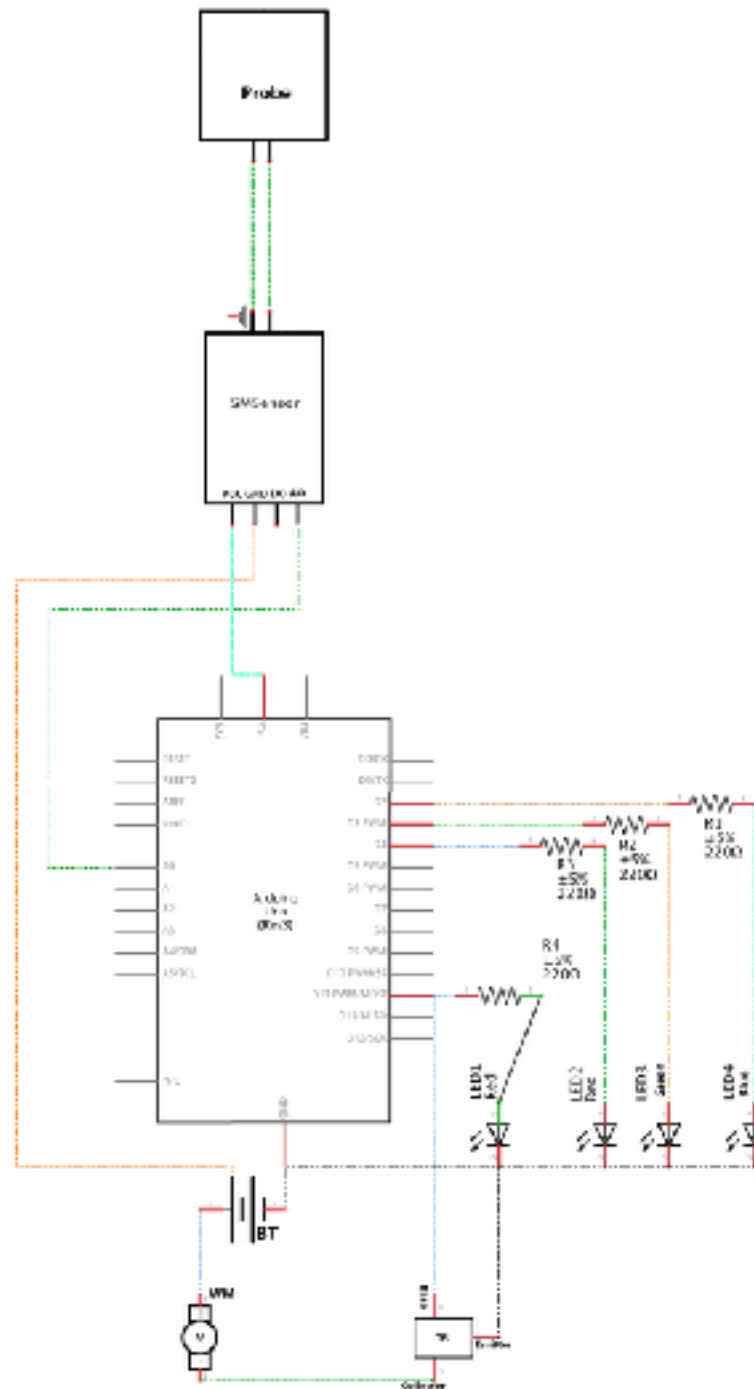
Komponentas	Žymėjimas	Paskirtis
LED raudonas	LED1	Informuoja apie įjungtą vandens variklį.
LED raudonas	LED2	Informuoja, kai vandens kiekis žemėje yra mažas.
LED žalias	LED3	Informuoja, kai vandens kiekis žemėje yra normalus.
LED mėlynas	LED4	Informuoja, kai vandens kiekis žemėje yra per didelis.
Rezistorius, 220 kΩ	R1	Skirtas LED1 raudonai.
Rezistorius, 220 kΩ	R2	Skirtas LED2 raudonai.
Rezistorius, 220 kΩ	R3	Skirtas LED3 žaliai.

Rezistorius, 220 kΩ	R4	Skirtas LED4 mėlynai.
Vandens variklis (pompa) 6V	WM	Tiekti vandenį į drėkinimo sistemą.
Tranzistorius IRF530	TR	Skirta vandens varikliui.
Dirvožemio drėgmės higrometro zondas (6cm x 2cm)	Probe	Įstaliuojamas į žemę, dirvožemio drėgmės higrometro dalis, matuojanti elektros varžą žemėje.
Dirvožemio drėgmės higrometro modulis	SMSensor	Matuoja elektros varžą žemėje, pateikia duomenys mikrovaldikliui.
Papildomas maitinimas	BT	Papildomas maitinimas vandens varikliui.
Adapteris	AC	Maitinti įrenginio sistemą.
Ultra violetinių spindulių šviesos diodai 12V	UV	LED, skirtas tiekti UV spindulius augalui. Šiuo metu nėra instaliuota į įrenginį.
Temperatūros modulis	TMP	Matuoja aplinkos temperatūrą. Šie duomenys bus naudojami nustatant, ar reikia įjungti vandens apipurškimą. Šiuo metu nėra instaliuota į įrenginį.
Servo motoriukas MG09R	SERVO	Vandens srovės perjungėjas, tarp vandens srovės apipurškimo bei drėkinimo sistemų. Šiuo metu nėra instaliuota į įrenginį.

## 2.2. Arduino komponentų sujungimo schema

Schematiškai pavaizduota, kaip projekte sujungti komponentai į veikiančią (praktiškai patikrintą) sistemą. Veikiančios sujungtos Arduino sistemos prototipo nuotrauka 1 priede.

**1 schema.** Projekte naudojamų Arduino komponentų sujungimo schema



### 2.3. Konstrukcinės dalies komponentų panaudojimas

Projektui įgyvendinti taip pat svarbūs konstrukciniai - mechaniniai sprendimai. Buitinių daiktų pritaikymas reikiamam funkcionalumui pasiekti. Panaudoti komponentai yra pateikti 2 lentelėje.

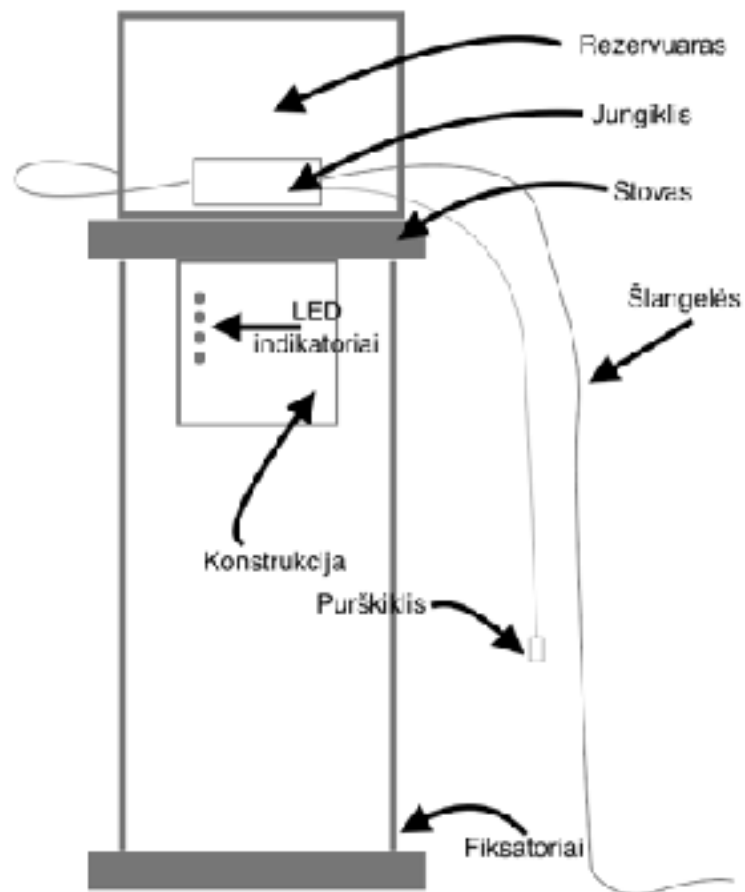
**2 lentelė.** Konstrukcinės - mechaninės dalies komponentai

Komponentas	Žymėjimas	Paskirtis
Batų stovas	Stovas	Naudojas kaip įrenginio stovas. Maksimalus ištraukto stovo ilgis yra 170 cm, tai leidžia stovą adaptuoti prie augalo aukščio.
5 litrų talpos kibiras	Rezervuaras	Vandens rezervuaras, skirtas laistyti augalą.
Metaliniai konstrukciniai elementai	Konstrukcija	Metalinės konstrukcinės plokštės, skirtos užtikrinti mechaninį sistemos stabilumą bei atliekančios įtaisų tvirtinimo funkciją.
Vandens šlangelės 3 metrai	Šlangelės	Vandens šlangelės skirtos drėkinti augalą. Šlangelė, drėkinantį žemę turi daug smulkių skylių, skirtai tolygiam vandens paskirstymui žemėje.
Vandens srovės perjungėjas	Jungiklis	Srovės perjungėjas valdo vandens padavimą tarp apipurškimo arba dreginimo vandens šlangelių.
Konstrukciniai fiksatoriai	Fiksatoriai	Fiksatoriai, skirti užtikrinti konstrukcijos stabilumą bei vandens nepralaidumą.
Antgalis purškimui	Purškiklis	Augalo apipurškėjas.

## 2.4. Konstrukcinės dalies schema

Schematiškai pavaizduota sukurta konstrukcija. Konstrukcinio prototipo nuotraukos 1, 2 prieduose.

2 **schema.** Konstrukcinė - mechaninė sujungimo schema



## 3. PROGRAMINĖ DALIS

### 3.1. Įgyvendintų klasių ir metodų aprašymas

Programinė dalis yra implemetuota objektiškai. Visiems pagrindiniams sukurto įrenginio Arduino platformos moduliams yra sukurtos atskiros klasės.



**Info** klasė skirta valdyti LED2, LED3, LED4 šviesos diodo indikatorius. Tokiu būdu, siekiama sukurti vienodą varototojo notifikavimo sistemą įrenginio veikimo metu. **Notification (int Led, int Times)** metodas siuncia atitinkamam šviesos diodui tam tikrą kiekį signalų. Ši metodą sudaro parametrai, skirti nurodyti priskirtai šviesos diodo konstantai (defined pin) bei mirgsėjimo kartams. **HardwareWarning ()** metodas skirtas notifikuoti vartotoją apie tai, kad nustojo veikti kuris nors iš įrenginio komponentų. Metodas, siunčia tris trumpus mirgtelėjimus į visus šviesos diodus.

**Hygro** klasė skirta tvarkyti duomenis, grįžtančius iš dirvožemio drėgmės jutiklio. **Ret\_values (int DelayTime)** - metodas apdoroja grįžtančią aibę duomenų iš jutiklio. Šiame metode naudojamas C++ MAP containeris, kuris paverčia grįžtančią duomenų aibę procentinę išraišką. Parametre nurodomas "užlaikymo" (delay) intervalas.

**WaterPomp** klasė skirta vandens pompoms valdymui. **DuneMode ()** metodas iššaukiamas, kai vandens kiekis žemėje yra mažas. **DuneMode ()** metode yra inicializuojamas **Hygro::ret\_values** metodas, tam, kad **DuneMode ()** metodas papildomai patikrintų, ar iš tikrųjų duomenys yra tinkami motoriuko paleidimui. Dirvožemio drėgmės jutiklio duomenys yra tikrinami 10 kartų (konstanta), prieš įjungiant motoriuką. Tai padaryta tam, jog gražinus jutikliui blogus duomenys (o tai labai retai pasitaiko), neįvyktų netinkama drėkinimo procedūra, kurios ilgalaikis rezultatas taptų augalo nykimas. Atitikus sausos žemės sąlygas, motoriukas yra leidžiamas 5 sekundes. Jei atlikus drėkinimą ir toliau neatitinka reikiamos drėgmės sąlygų, ciklas yra kartojamas po kelių sekundžių, priešingu atveju iš metodo grįžtama į pagrindinį ciklą. Taipogi metodas turi patikrinimą, skirta nustatyti ar dirvožemio jutiklis yra iš tikrųjų prijungtas.

**SelfCheckDigit()** ir **SelfCheckAnalog()** funkcijos, skirtos sistemos diagnostikai. Programos vykdytomo pradžioje, yra kviečiami visi įrenginio moduliai, tokiu būdu įsitikinama, kad nėra sisteminio gedimo. Programos kodas yra pateikiamas 4 priede.

## 4. ĮRENGINIO TESTAVIMAS, DEFEKTŲ ŠALINIMAS

Įrenginio testavimas sudarė du etapus: pirminio testavimo ir testavimo, kai įrenginys pradėjo atlikti augalo priežiūrą.

Rasti pirminio testavimo defektai:

- Defektai, susiję su įrenginio mechaninių komponentų pralaidumu vandeniui. Nuotrauka pirminio įrenginio testavimo 3 priede.
- Paaiškėjo, kad turimas servo motoriukas (SG90) yra per silpnas (1kg/cm), pasukti vandens srovės perjungėją. Dėl to, silpnas motoriukas iš dabartinės konstrukcijos buvo pašalintas. Šiuo metu užsakytas daug stipresnis (13 kg/cm) servo motoriukas (MG09R), kuris turėtų susidoroti su užduotimi. Laikinas sprendimas - vandens perjungimo sklendė pastatyta taip, kad vykdant augalo drėkinimą, dalis vandens būtų purškiama ant šakų, kita dalis drėkintų dirvožemį.

Defektai kai įrenginys pradėjo atlikti augalo priežiūrą:

- Kai įrenginys buvo paliktas parai laiko be papildomas priežiūros, visas rezervuaro turinys (4 litrai vandens) buvo užpildytas ant eglutės, išsiliejo ant grindų. Šios avarijos kaltininkas buvo musu ofiso katinas, kuris nuplėšė kontaktus nuo dirvožemio drėgmės jutiklio. Jutiklis pradėjo grąžinti pastoviai 0 reikšmę, rezultatas atitiko metode **WaterPomp::DuneMode()** keliamas sąlygas motoriukui paleisti. Po šio incidento, programiniame kode buvo sudėti pagrindiniam cikle bei pačiam **DuneMode()** metode patikrinimai, kad panašūs incidentai ateityje nepasikartotų.
- Po pusantros savaitės, drėgmės jutiklis nustojo grąžinti teisingus duomenys. Šios problemos priežastis buvo nuplauti dirvožemio drėgmės higrometro zondo kontaktai. Priežastis nekokybiški (pigūs) kitiški sensoriai. Elektros varžai matuoti buvo pritaikyti du metalo laidininkai, prie kurių prilituoti laidai einantys į dirvožemio drėgmės higrometro modulį - jie puikiai pakeitė nekokybišką daviklį.

## 5. ĮRENGINIO PLĖTOTĖ

Įrenginys nėra baigtinis ir bus plėtojamas, norint pasiekti numatytam funkcionalumui, šiuo metu trūksta:

- Servo motoriuko MG09R, kuris skirstytų vandens padavimą tarp apipurškimo arba drėkinimo;
- 50 cm 12V pilno spektro UV LED apšvietimo juostos, pakankamam augalo UV apšvietimui užtikrinti;
- Programinio kodo, skirta implementuoti šių modulių funkcionalumui.

Abudu komponentai yra užsakyti, laukiama, kol bus pristatyti.

# IŠVADOS

Arduino platforma atveria plačias galimybes realizuoti savo kūrybines idėjas. Praėjo laikai, kai AVR mikroschemas programuodavo tik tai tos srities specialistai. Tuo metu didžiausia problema buvo ta, kad programuoti nebuvo nei lengva nei patogiu, nes programų kodas buvo panašus į assemblerį. Arduino platforma visa tai paslėpė po gražiais metodų pavadinimais, o tai padarė patį kūrimo procesą šioje platformoje suprantamą ir įdomų paprastam techniniam vartotojui. Susipažinus su baziniais veikimo principais ir turint programavimo žinių, galima nesunkiai automatizuoti nemažai buityje vykdomų procesų, kurie taupo žmogaus laiką.

Atlikus įrenginio testavimą bei pašalinus visus rastus defektus įrenginys tapo visiškai funkcionuojantis. Taigi, mokomosios praktikos tikslas yra pasiektas - augalų prižiūri sukurtas automatinis augalo auginimo įrenginys. Augalo drėkinimo procesas tapo visiškai automatizuotas ir pakankamai autonominis, kadangi augalo prižiūrėtoji teliko rūpintis vandens pripildymu sistemoje.

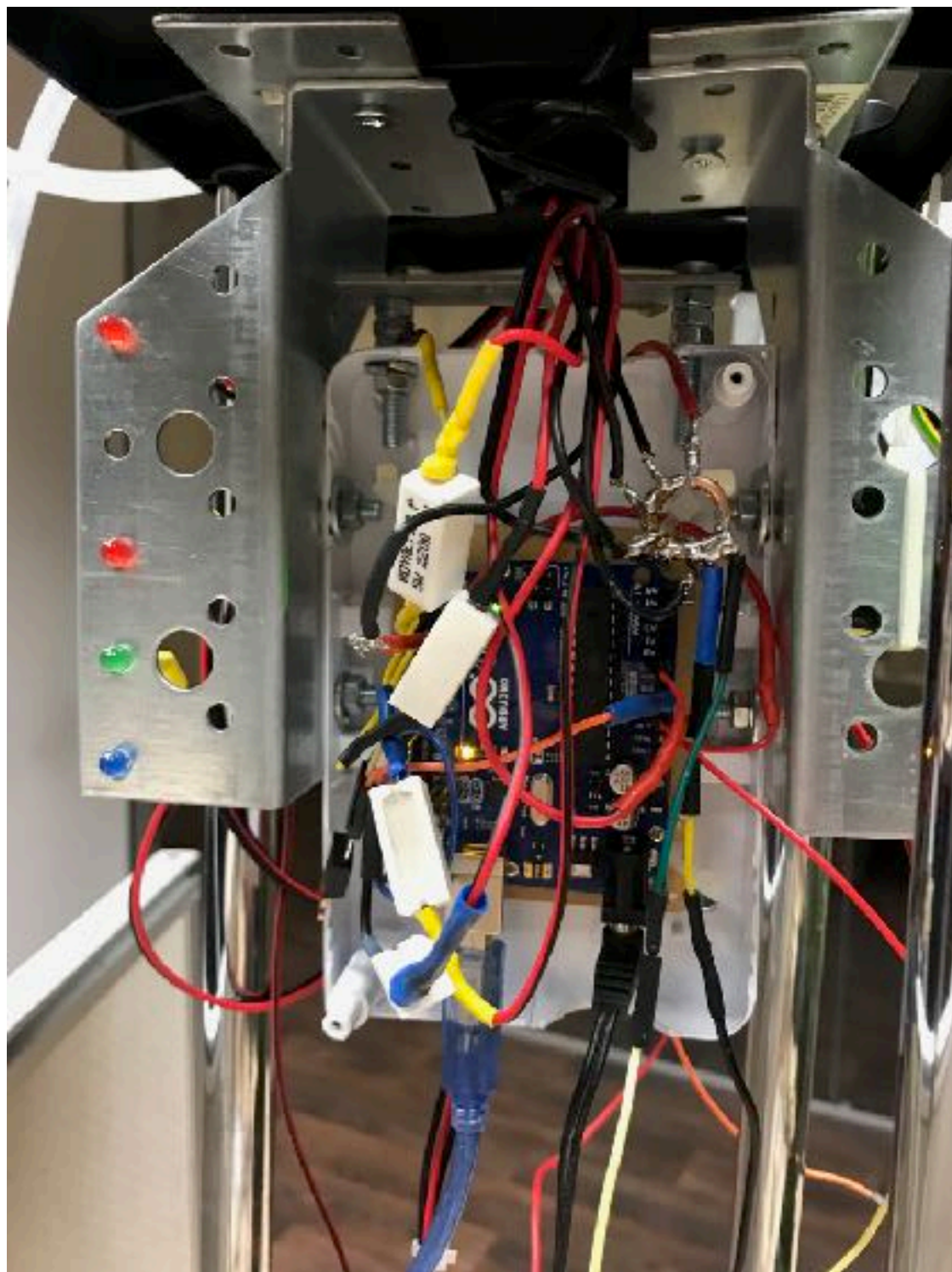
Mokomosios praktikos metu, įgyta patirtis gali padėti kuriant panašias automatizuotas sistemas nekartojant tų pačių kūrimo metu pasitaikiusių klaidų.

## LITERATŪROS SĄRAŠAS

1. Thomas Hylland Eriksen. Akimirkos tironija. Vilnius: Tyto alba, 2004.
2. Irmantas Girdauskas. Baigiamasis magistro projektas: Arduino mikrovaldiklio plokštės panaudojimo procesų valdymui galimybių tyrimas. Kaunas: KTU, 2016. Prieiga per internetą: <https://core.ac.uk/download/pdf/51552104.pdf>.
3. Edita Jakubauskaitė - Turskienė. Kambarinė eglutė aukštoji araukarija. Prieiga per internetą: <http://www.manonamai.lt/mano-sodas-ir-kiemas/kiemas/kambarine-eglute-aukstoji-araukarija.d?id=73472610>.

## **PRIEDAI**

*1 priedas. Sujungtos Arduino sistemos prototipo nuotrauka*



*2 priedas. Konstrukcinio prototipo nuotrauka*





### *3 priedas. Pirminis įrenginio testavimas*





#### *4 priedas. Pirminis įrenginio testavimas*

```
#define WATERMOTOR 11
#define ALERT0 2
#define ALERT1 3
#define ALERT2 4
#define TEMPERATURE 5
#define HYGROMETER A0

boolean SystemCheck = false;
boolean TetMode = false;
int MaxWater = 90;
int MinWater = 40;
int CheckTime = 1000;

void setup() {

  pinMode(HYGROMETER, INPUT);
  pinMode(WATERMOTOR, OUTPUT);

  pinMode(ALERT0, OUTPUT);
  pinMode(ALERT1, OUTPUT);
  pinMode(ALERT2, OUTPUT);

  Serial.begin(9600);
}

class Info {

private:
  int NotDelay = 100;
  int HardTimes = 10;
  int HardDelay = 50;

public:
  void Notification(int, int);
  void HardwareWarning();
};

void Info::Notification (int Led, int Times) {

  for (int i=0; i<Times; i++){

    for (int k=1; k>-1; k--){
```

```

        digitalWrite(Led, k);
        delay(NotDelay);

    }
    delay(NotDelay);

}

}

void Info::HardwareWarning () {

    int i = 0;

    while(HardTimes>i){

        int h = 1;

        do{

            digitalWrite(ALERT0, h);
            digitalWrite(ALERT1, h);
            digitalWrite(ALERT2, h);
            delay(HardDelay);
            h--;

        }while(h>-1);

        i++;

    };

}

class Hygro {

private:
    Info inf;
    short MaxVal = 0;
    short MinValue = 940;
    short MaxPerc = 100;
    short output_value ;

public:
    int ret_values (int);

};

```

```

int Hygro::ret_values (int DelayTime) {

    output_value= analogRead(HYGROMETER);
    output_value = map(output_value,MinValue,MaxVal,0,MaxPerc);
    delay(DelayTime);

    return output_value;
}

```

```

class WaterPomp {

private:
    Hygro dry;
    Info info;
    int CheckTimes = 10;
    int DelayCheckTimes = 1000;
    int RunningMotorTime = 5000;

public:
    void DuneMode();

};

```

```

void WaterPomp::DuneMode () {

    int HygroVal = 0;
    int i = 0;

    repeat:
    boolean ActivateMotor = true;

    do{
        delay(DelayCheckTimes);
        int HygroVal = dry.ret_values(1);
        Serial.println(HygroVal);

        if (HygroVal<=0){
            info.HardwareWarning();
            ActivateMotor = false;
            break;
        }
        else{

            if (HygroVal<=MinWater){
                info.Notification(ALERT0, 4);
                i++;
            }
            else{
                ActivateMotor = false;
                info.Notification(ALERT1, 4);
            }
        }
    }
}

```

```

        break;
    }

}

} while(i<CheckTimes);

if (ActivateMotor){
    digitalWrite(WATERMOTOR, HIGH);
    delay(RunningMotorTime);
    digitalWrite(WATERMOTOR, LOW);
    i = i-2;
    goto repeat;
}

}

Info info;
Hygro hygro;
WaterPomp pump;

void SelfCheckDigit(){

    int Modules [] = {WATERMOTOR, ALERT0, ALERT1, ALERT2};
    int l = (sizeof(Modules)/sizeof(*Modules));

    for (int i=0;i<l;i++){

        int z = 1;

        do{

            digitalWrite(Modules[i], z);
            delay(1000);
            z--;

        } while(z>-1);

    }

    info.HardwareWarning();

}

void SelfCheckAnalog(){

```

```

int AnalogCheckDelay = 1000;
int TimeToCheck = 10;

delay(AnalogCheckDelay);

for (int i=0; i<TimeToCheck; i++){

    delay(AnalogCheckDelay);

    if (hygro.ret_values(0)>0){
        info.Notification(ALERT1, 2);
        break;
    }
    else if (i==TimeToCheck-1){

        info.Notification(ALERT0, 100);
    }

}

info.HardwareWarning();
}

void loop() {

    if (SystemCheck){
        SelfCheckDigit();
        SelfCheckAnalog();
        SystemCheck = false;
    }

    Stop:

    int HygroVal = hygro.ret_values(CheckTime);
    Serial.println(HygroVal);

    if (!TetMode){

        if (HygroVal<=0){
            info.HardwareWarning();
            goto Stop;
        }

        if (HygroVal>=MaxWater){
            info.Notification(ALERT2, 2);

```

```
    }  
    else if (HygroVal>MinWater){  
        info.Notification(ALERT1, 2);  
    }  
    else if (HygroVal<=MinWater){  
        pump.DuneMode();  
        info.Notification(ALERT0, 2);  
    }  
  
    Serial.println(HygroVal);  
}  
  
}
```