



VILNIAUS VERSLO KOLEGIJA

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

**AUGALO AUGINIMO AUTOMATIZAVIMAS NAUDOJANT
ARDUINO PLATFORMĄ**

MOKOMOSIOS PRAKTIKOS ATASKAITA

Darbą atliko: **Povilas Svetovas**

Darbo vadovas: lekt. **Aivaras Čechamiras**

Darbas originalus _____ Povilas Svetovas
(studento parašas)

Vilnius, 2018

TURINYS

| | |
|--|-----------|
| ĮVADAS | 3 |
| 1. ĮRENGINIO FUNKCINIS APRAŠYMAS | 5 |
| 2. TECHNINĖ ĮRENGINIO REALIZACIJA | 6 |
| 2.1. Arduino komponentų aprašymas | 6 |
| 2.2. Arduino komponentų sujungimo schema | 7 |
| 2.3. Konstrukcinės dalies komponentų panaudojimas | 8 |
| 2.4. Konstrukcinės dalies schema | 8 |
| 3. PROGRAMINĖ DALIS | 10 |
| 3.1. Klasių ir metodų aprašymas | 10 |
| 4. ĮRENGINIO TESTAVIMAS, DEFEKTŲ ŠALINIMAS | 11 |
| 5. ĮRENGINIO PLĖTOTĖ | 12 |
| IŠVADOS | 13 |
| LITERATŪROS SĄRAŠAS | 14 |
| PRIEDAI | 15 |

IVADAS

Darbo aktualumas. Šiuolaikinio gyvenimo tempas žmogui sukelia daug iššūkių. Vienas iš garsiausių šių laikų sociologų Thomas Hylland Eriksen 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. Nesiremiami 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 įnoringas ir lepus augalas. Ši eglutė nemėgsta daug saulės spindulių, o laistyti aurakariją reikia saikingai, o tai reiškia, kad žemė vazone turi būti drėgna, bet ne šlapia. Svarbu augalą apipurkšti bent keletą kartų per dieną (Jakubauskaitė, 2017). Tokias sąlygas yra ypač sunku užtikrinti biuro patalpose.

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

2. TECHNINĖ ĮRENGINIO REALIZACIJA

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

2.1. Arduino komponentų aprašymas

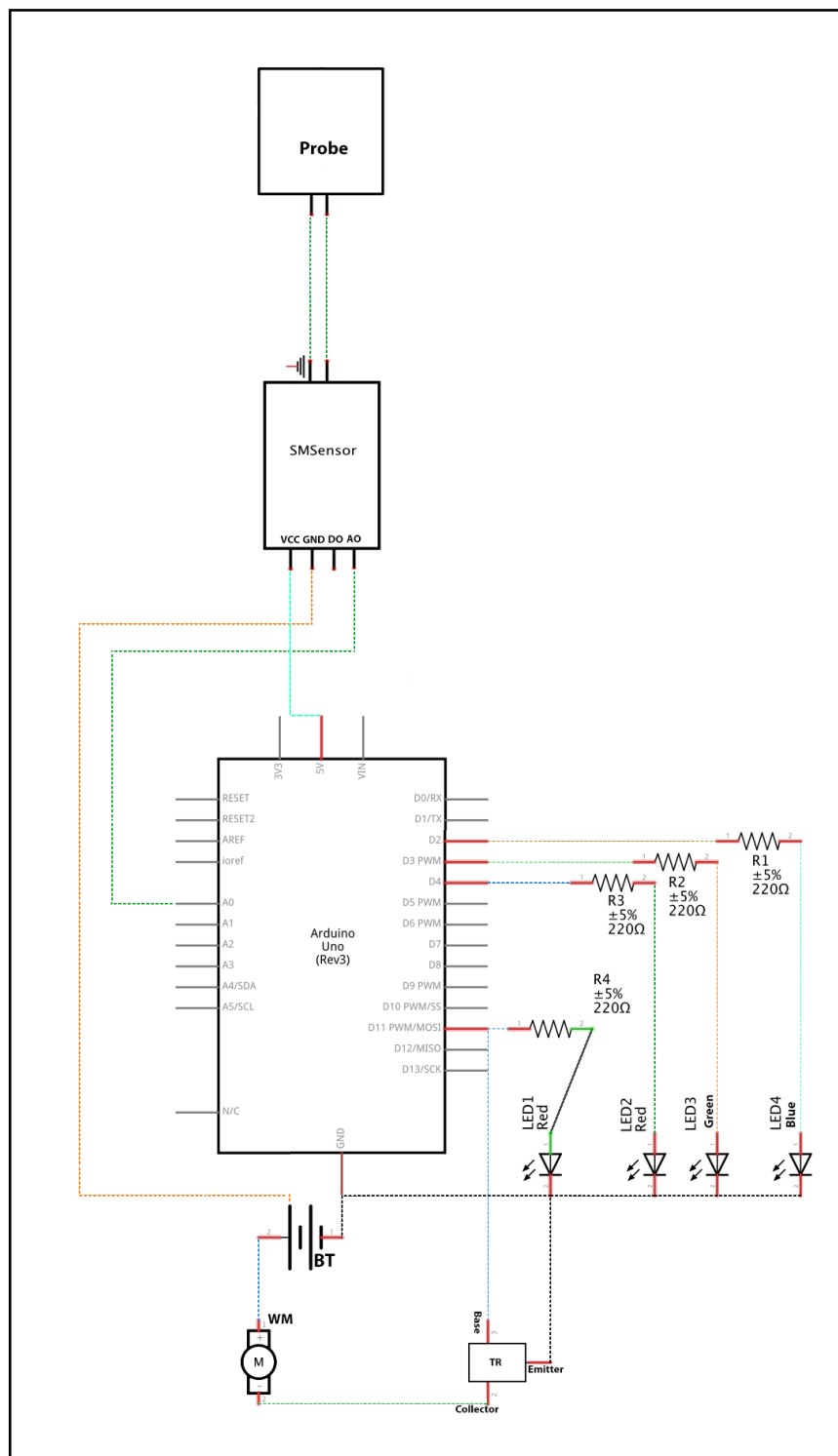
Projekte naudojama Arduino UNO R3 mikrovaldiklio plokštė, turinti instaliuotą 8-bitų ATmega 328 mikrovaldiklį. Taip pat naudojami ir 1 lentelėje nurodyti komponentai.

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

Schemoje žemiau (1 pav.) yra pavaizduota, kaip projekte sujungti komponentai į veikiančią (praktiškai patikrintą) sistemą. Veikiančios sujungtos Arduino sistemos prototipo nuotrauka pateikta 1 priede.



1 pav. Projekte naudojamų Arduino komponentų sujungimo schema

2.3.Konstrukcinės dalies komponentų panaudojimas

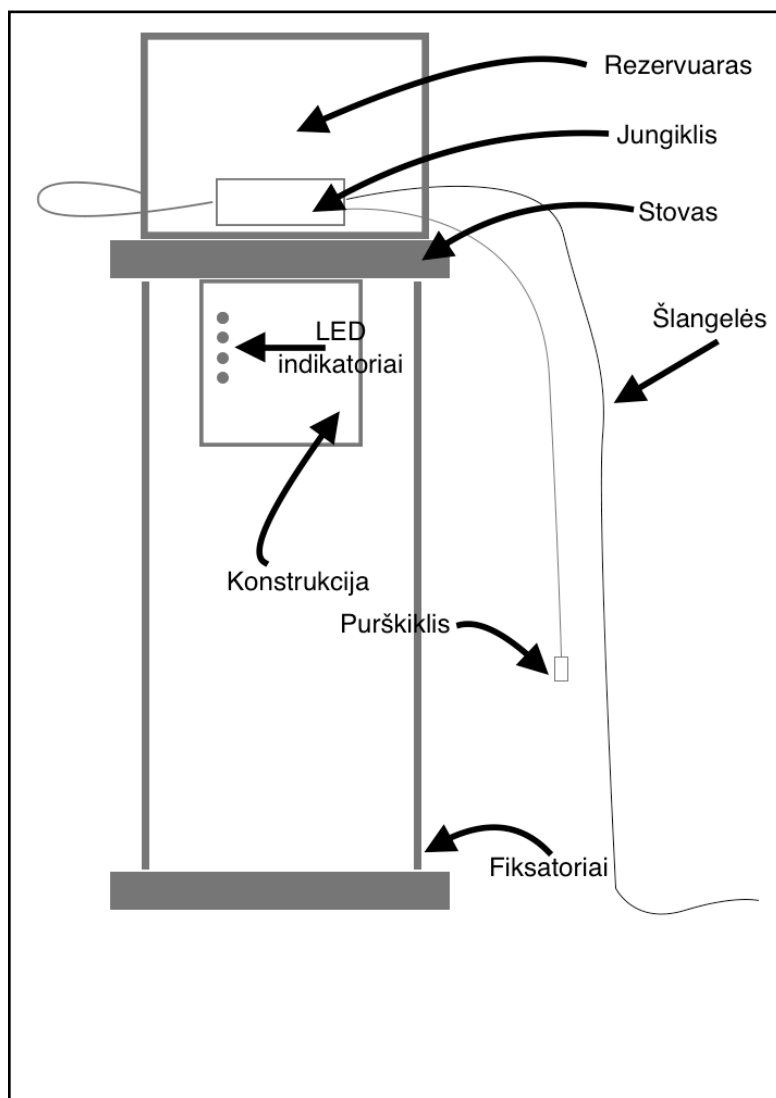
Projektui įgyvendinti taip pat svarbūs konstrukciniai-mechaniniai sprendimai. 2 lentelėje pateikiami buityje naudojami daiktai, kurie buvo panaudoti projekte, norimam funkcionalumui pasiekti (2 lentelė).

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 skylučių, 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

Schemoje žemiau pavaizduota sukurta konstrukcija (2 pav.). Konstrukcinio prototipo nuotraukos pateikiamos 1 ir 2 prieduose.



2 pav. Konstrukcinė-mechaninė sujungimo schema

3. PROGRAMINĖ DALIS

3.1. Klasių ir metodų aprašymas

Programinė dalis yra implemetuota objektiškai. Visiems pagrindiniams sukurtu 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 siunčia 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. Šis metodas, siunčia tris trumpus mirgtelejimus 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 konteineris, kuris paverčia grįžtančią duomenų aibę procentine išraiška. Parametre nurodomas laiko užlaikymo (delay) intervalas.

WaterPomp klasė skirta vandens pompos 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. Taip padaryta tam, kad grąžinus jutikliui nekorektiškus duomenis (o tai labai retai pasitaiko), neįvyktų nereikalinga drėkinimo procedūra.

Jeigu žemė sausa, motoriukas yra leidžiamas 5 sekundes, tuo metu žemė drėkinama. Jei po drėkinimo žemės drėgnumas neatitinka deklaruotos *MinWater* konstantos, ciklas yra kartojamas po kelių sekundžių, priešingu atveju iš *DuneMode()* metodo grįžtama į pagrindinį ciklą. Taip pat šis metodas turi patikrinimą, skirta nustatyti ar dirvožemio jutiklis yra iš tikrųjų prijungtas.

SelfCheckDigit() ir *SelfCheckAnalog()* funkcijos yra skirtos sistemos diagnostikai. Programos vykdymo 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 pirminiai testavimo defektai:

1. Defektai, susiję su įrenginio mechaninių komponentų pralaidumu vandeniui.
2. Paaiškėjo, kad turimas Servo motorius (SG90) yra per silpnas (1kg/cm), pasukti vandens srovės perjungėją. Dėl to, silpnas motorius iš dabartinės konstrukcijos buvo pašalintas. Šiuo metu užsakytas daug stipresnis (13 kg/cm) Servo motorius (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 (3 priedas).

Defektai, kai įrenginys pradėjo atlikti augalo priežiūrą:

1. Kai įrenginys buvo paliktas parai laiko be papildomos priežiūros, visas rezervuaro turinys (4 litrai vandens) buvo užpildytas ant eglutės, išsiliejo ant grindų. Šios avarijos kaltininkas buvo mūsų ofiso katinas, kuris nuplėšė kontaktus nuo dirvožemio drėgmės jutiklio. Jutiklis pradėjo grąžinti pastoviai 0 reikšmę, rezultatas atitiko *WaterPomp::DuneMode()* metode keliamas sąlygas motoriumui paleisti. Po šio incidento, programiniame kode pagrindiniame cikle bei pačiame *DuneMode()* metode buvo sudėti patikrinimai, kad ateityje nepasikartotų panašūs incidentai.
2. Po pusantros savaitės, drėgmės jutiklis sugedo ir pradėjo rodyti nekorektiškus duomenis. Gedimo priežastis – laistomas vanduo nuplovė dirvožemio drėgmės higrometro zondo kontaktus. O to priežastis – nekokybiški (pigūs) 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 užbaigtas. Jis toliau bus tobulinamas, norint pasiekti numatytą funkcionalumą. Šiuo metu trūksta:

1. Servo motoriuko MG09R, kuris skirstytų vandens padavimą tarp šakų apipurškimo arba žemės drėkinimo;
2. 50 cm 12V pilno spektro UV LED apšvietimo juostos, pakankamam augalo UV apšvietimui užtikrinti;
3. Programinio kodo, skirto implementuoti šių modulių funkcionalumui.

Abu komponentai yra užsakyti, laukiama, kol bus pristatyti.

IŠVADOS

1. Arduino platforma atveria plačias galimybes realizuoti savo kūrybines idėjas. Praėjo laikai, kai AVR mikroschemas programuodavo tik tos srities specialistai, nes programų kodas buvo panašus į assemblerį. Arduino platforma kūrimo procesą padarė suprantamą ir įdomų paprastam techniniam vartotojui. Susipažinus su baziniais veikimo principais ir turint programavimo žinių, galima nesunkiai automatizuoti nemažai buityje vykdomų ir daug laiko atimančių procesų.
2. 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 automatizuotas, o augalo šeimininkui liko pripildyti vandens rezervuarą.
3. Mokomosios praktikos metu įgyta patirtis gali padėti kuriant panašias automatizuotas sistemas nekartojant tų pačių kūrimo metu pasitaikiusių klaidų.

LITERATŪROS SĄRAŠAS

1. Eriksen, T. H. *Akimirkos tironija*. Vilnius: Tyto alba, 2004.
2. Girdauskas, I. 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> [žiūrėta: 2017-03-01].
3. Jakubauskaitė – Turskienė, E. *Kambarinė eglutė aukštoji araukarija*. Prieiga per internetą: <http://www.manonamai.lt/mano-sodas-ir-kiemas/kiemas/kambarine-eglute-aukstoji-araukarija.d?id=73472610> [žiūrėta: 2017-03-01].

PRIEDAI

1 priedas. Sujungtos Arduino sistemos prototipo nuotrauka







```
#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, AIERT1, AIERT2};
    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);

}

}

```