

TARTU ÜLIKOOL
Arvutiteaduse instituut
informaatika õppekava

Selena Lubi

Arduino stardikomplekti õppematerjalid

Bakalaureusetöö (9 EAP)

Juhendajad: Alo Peets, Taavi Duvin, Anne Villems

Tartu 2016

Arduino stardikomplekti õppematerjalid

Lühikokkuvõte:

Käesoleva bakalaureusetöö eesmärgiks on tutvustada Arduinoga ühilduvate andurite ja lisatarvikute kasutusvõimalusi ning koostada neist huvitavamate kohta eestikeelsed õppematerjalid. Töö on jaotatud kolmeks osaks. Esimeses peatükis antakse ülevaade Arduino platvormist ja stardikomplekti olemusest. Teises peatükis tutvustatakse stardikomplekti komponentide kasutusvõimalusi ning koostatakse juhend arenduskeskkonna seadistamiseks. Töö kolmandas selgitatakse andurite praktilist kasutamist 10 kasvava raskusastmega näidisprojekti abil.

Võtmesõnad: Arduino, stardikomplekt, robotika

CERCS: P170(Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine)

Arduino learning materials based on starter kit

Abstract:

The aim of this bachelor thesis is to introduce sensors and extra components that can be used with Arduino and create learning materials based on some of them. Materials are in Estonian and using most interesting components of starter kit. The thesis is divided into three parts. Chapter 1 introduces Arduino platform and purpose of starter kit. Chapter 2 describes sensors and extra components that are included in starter kit and shows how to set up Arduino IDE. Chapter 3 explains how to use some of the sensors and components in practice by giving 10 example projects with growing level of difficulty.

Keywords: Arduino, starter kit, robotics

CERCS: P170(Computer science, numerical analysis, systems, control)

Sisukord

Sissejuhatus	5
1 Arduino komplektide võimalused.....	6
1.1 Platvormi tutvustus	6
1.2 Arduino ajalugu ja populaarsus	6
1.3 Teiste tootjate Arduinol põhinevad plaadid	8
1.4 Erinevad Arduino tooted.....	9
1.4.1 Arduino arendusplaadid.....	9
1.4.2 Laiendusplaadid.....	14
1.4.3 Arduinol põhinevad komplektid.....	15
1.5 Arduino programmeerimine.....	17
1.6 Võrdlus Raspberry Pi-ga.....	18
2 Stardikomplekti sisu ja seadistamine.....	20
2.1 Stardikomplekti sisu.....	20
2.1.1 Kõikides projektides kasutatavad komponendid	23
2.1.2 Valgusdiodid.....	23
2.1.3 Takistid	24
2.1.4 Lülitid	25
2.1.5 Projekti juhtimiseks vajalikud tarvikud	25
2.1.6 Mootorid	27
2.1.7 Erkaanid ja näidikud	28
2.1.8 Keskkonnale reageerivad andurid.....	29
2.1.9 Teised komponendid.....	31
2.2 Arenduskeskkonna paigaldamine ja arendusplaadi ühendamine.....	35
2.3 Koodi kirjutamine ja plaadile laadimine	39
2.4 Maketeerimislaua ja protoplaadi kasutamine.....	41
3 Õppematerjalid	43
3.1 Erinevad võimalused LED vilgutamiseks ja FOR-tükkel.....	43
3.2 Lüliti ühendamine ja LEDi reageerimine vastavalt lüliti asendile vooluringis ja digitaalsignaali abil IF-lausega	46
3.3 RGB ledi ühendamine ja RGB koodide katsetamine analogWrite abil	50
3.4 Kolme nupu abil RGB LEDi lülitamine	53
3.5 Ekraani ühendamine ja teksti kuvamine 1602 LCD ekraanil	55
3.6 Helianduri ühendamine, LEDide vilgutamine vastavalt müratugevusele.....	59

3.7	Fototakisti abil valguse mõõtmine, Serial monitoriga tutvumine, valgustundlik LED	62
3.8	Juhtkangi ühendamine ja mootorite juhtimine.....	65
3.9	Infrapuna vastuvõtja, pult ja 7-segmendiline ühe numbriga LED	69
3.10	Temperatuuri kuvamine 4-kohalisel 7-segmendilisel näidikul	74
	Kokkuvõte.....	81
	Kasutatud kirjandus	82
	Litsents.....	87

Sissejuhatus

Tänapäeva ühiskonnas on tehnika omandanud asendamatu rolli, kuid tööjõuturul valitseb IT-spetsalistide puudus. Tihti pole põhikooli ja gümnaasiumi lõpetajatel selget ettekujutust, mida robotika, programmeerimine ja IT valdkond endast täpsemalt kujutavad. Reaalained on üldhariduskoolides pigem teoreetilise suunitlusega, mis jätab õpilastele kogu tehnika ja IT valdkonnast keerulise ning igava mulje. Seepärast otsustatakse ka edasistes õpingutes millegi tuttavama ja turvalisema kasuks.

Haridusasutuste huvi programmeerimise ja robotika õpetamise vastu on olemas. Paraku on sobivate materjalide ja vastavate teadmistega õpetajate puuduse ning robotikakomponentide kõrge hinna tõttu informaatika ja robotika õpetamine paljudes eesti koolides veel üsna madalal tasemel. Koolidel on võimalus ise otsustada, kuidas infotehnoloogiat õpetada, mille tõttu puuduvad ka ühtsed õpetamismeetodid ning üldlevinud õppematerjalid [1].

Käesoleva bakalaureusetöö eesmärk on tutvustada Arduinoga kasutatavaid komponente ja andureid, mis sisalduvad umbes 50 eurot maksvas Infduino stardikomplektis ning koostada neist huvitavamate kohta eestikeelsed õppematerjalid.

Töö on jaotatud kolme peatüki vahel. Esimeses peatükis antakse ülevaade Arduino platvormist ja stardikomplekti olemusest. Teises peatükis tutvustatakse stardikomplekti komponentide kasutusvõimalusi ning koostatakse juhend arenduskeskkonna seadistamiseks. Töö kolmandas peatükis selgitatakse andurite praktilist kasutamist kümne kasvava raskusastmega näidisprojekti abil.

1 Arduino komplektide võimalused

Käesolevas peatükis tutvustatakse Arduino arendusplaate, võrreldakse erinevaid Arduino komplekte, nende kasutusvõimalusi ja ajalugu. Tuuakse põhjendusi, miks antud platvorm on sobilik koolirobootikas kasutamiseks ning milliseid projekte on sellega võimalik teha. Kirjeldatakse ka Arduino eeliseid ja puudusi võrreldes teiste platvormidega. Samuti tutvustatakse Arduino programmide kirjutamiseks enim kasutatavat tarkvara - Arduino arenduskeskkonda.

1.1 Platvormi tutvustus

Arduino on avatud lähtekoodiga platvorm, mis põhineb kooliõpilaste jaoks kergesti õpitaval riistvaral ja Arduino IDE tarkvaral [2]. Riistvaraks on arendusplaat, mis koosneb mikrokontrollerist koos ühilduvuse jaoks vajalike komponentidega. Arendusplaat on saadaval kümneid eri mudelid, erinevate eesmärkide ja raskusastmetega projektide koostamiseks. Arendusplaadile laaditakse kood ning külge on võimalik ühendada laiendusplaate, andureid ja lisatarvikuid [3]. Täpsema ülevaate turul hetkel populaarsetest arendusplaatidest annab punkt 1.4.1.

Tarkvara poole pealt on välja töötatud C keelega sarnane Arduino programmeerimiskeel ja tasuta saadavalolev programmeerimist lihtsustav arenduskeskkond [2]. Üldise ülevaate Arduino programmeerimisest leiab punktist 1.5.

1.2 Arduino ajalugu ja populaarsus

Arduino asutati 2005. aastal Itaalias Ivera linnas ning firma põhiidee sai alguse vajadusest piisava võimsuse ja mõistlikku hinnaga arendusplaadi järele, mille abil õpilased saaksid luua elektroonikaprojekte. Arduino arendamise põhitiimi kuulusid David Cuartielles, Gianluca Martino, Tom Igoe, David Mellis ja Massimo Banzi [4].

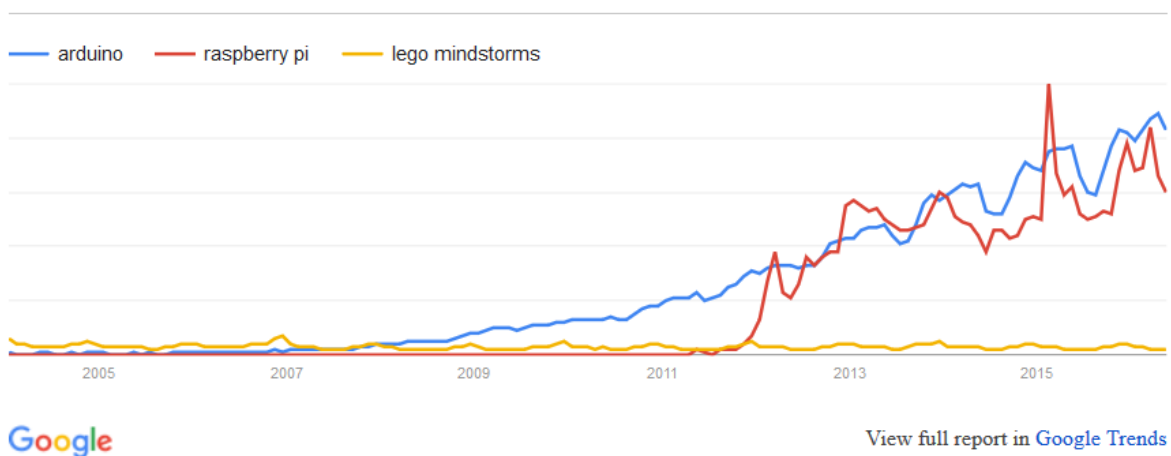
Ivrea interaktsioonidisaini instituudis (*Interaction Design Institute Ivrea* ehk IDII), kus üks Arduino loojatest, M. Banzi, õppijõuna töötas, oli sel hetkel kasutusel mikrokontroller BASIC Stamp, mis oli selleks hetkeks juba pea kümnendi kasutusel olnud ning jäi ka hinna

poolest õpilastele üsna kalliks. Põhiline puudus seisnes siiski selles, et BASIC Stamp-i jõudlusest jäi soovitud mõtete ellu viimiseks väheks. See ajendaski Banzi võtmaks ühendust kolleegiga Massachusettsi Tehnoloogiainstituudist (*Massachusetts Institute of Technology*), kellel oli kogemus disainerisõbraliku programmeerimiskeele Processing arendamisega, tekkis idee luua sarnane tarkvara ka mikrokontrollerite programmeerimiseks [4].

Üks IDII tudengitest, Hernando Barragán, alustas soovitud platvormi arendamisega, tulemuseks oli Wiring, mis sisaldas nii arendusplaati kui ka arenduskeskkonda [5]. Banzi võttis Wiringist küll eeskujuna, kuid Arduino lõpplahendus oli veelgi intuitiivsem ja odavam, makstes ilmumise hetkel 30 dollarit. Kuna eesmärk oli pakkuda odavat, kiiresti ja võimalikult paljude inimesteni jõudvat toodet, siis otsustati avatud lähtekoodi kasuks [4]. Riistvara on litsentsitud *Creative Commons* *Share-Alike 2.5* litsentsiga, mis lubab kasutamise nii isiklikel kui ka kaubanduslikel eesmärkidel, tingimusel, et järgi tehtud toode omab vabavaraalist litsentsi ning on märgitud, et eeskujuna on võetud Arduinost [6].

Innovaatiline lahendus jõudis kiiresti massidesse ning 2011 aastaks oli maailmas müüdnud juba üle 250 000 arendusplaadi. Täpseid müüginumbreid on hetkel raske öelda, kuna vabavaraalise riistvara tõttu on müügil palju Arduinol põhinevaid arendusplaate [7]. Inimeste huvist Arduino vastu annab aimu Google otsingutulemuste statistika aastast 2004 kuni mai 2016 (Joonis 1). Graafikul on võrdluseks välja toodud ka koolirobootika peamiste konkurentide Raspberry Pi ja Lego Mindstorms otsinute statistika.

Interest over time. Web Search. Worldwide, 2004 - present.



Google

[View full report in Google Trends](#)

Joonis 1. Google Otsingutulemuste statistika Arduino kohta [8]

Joonisel kujutatud info põhjal võib järeldada, et Arduino ei olnud vaid moetoode ning selle populaarsus maailmas jätkuvalt kasvab. Graafikul olevad langused jäävad suvekuudele, kui

enamustes maailma koolides on vaheaeg, mis omakorda näitab haridusasutuste tõusvat huvi robotika õppetöösse kaasamise vastu.

1.3 Teiste tootjate Arduinol põhinevad plaadid

Genuino on Arduino tütarfirma, mis loodi Arduino asutajate Massimo Banzi, David Cuartielles, Tom Igoe ja David Mellis poolt. Uue kaubamärgi loomise põhjuseks oli soov laiendada Arduino kaubamärki ka Euroopa turule, aga selgus, et üks asutajatest, Gianluca Martino, oli Arduino kaubamärgi juba Itaalias registreerinud. Seega otsustasid teised originaalliikmed Euroopas tutustatavaid plaate Genuino nime all müüma hakata [9].

Gianluca Martino lõpetas igasuguse koostöö ülejäänud algse tiimi liikmetega ning tema firma tegutseb lehel <http://www.arduino.org/>, teistele liikmetele valmistab pahameelt, et inimeste teadmatus tõttu saab G.Martino firma tunnustust nende loodud aktiivse kasutajatoega kogukonna eest, mis asub <https://www.arduino.cc/> [9].

Plaadid on komponentide ja omaduste poolest identsed Arduinole, ainsaks erinevuseks ongi kaubamärgi nimi ja turunduspiirkond. Nimelt turustatakse Genuinot Euroopas, Aasias, Lõuna- Ameerikas, Aafrikas ja Kanadas, seevastu Arduino peamiseks sihtturuks on Ameerika Ühendriigid [10].



Joonis 2. Arduino UNO ja Genuino UNO arendusplaadid [11]

Lisaks Genuinole, mille kvaliteet on võrdväärne Arduinoga, on müügil ka erinevate tootjate Arduinoga tarkvaraga ühilduvaid plaate. Mõned neist on oma plaadi arendamisel võtnud

Arduinost eeskju ning lisanud ka omaltpoolt erinevaid funktsionaalsusi - selliseid plaate nimetatakse derivaatideks. Samas on ka neid tootjaid, kes ise lisafunktsionaalsust lisamata kopeerivad Arduino lahendusi, kuid omavad vähemalt oma kaubamärki – neid kutsutakse kloonideks. Paraku ringleb turul ka järgi tehtud Arduino märgisega tooteid ehk võltsinguid. Avatud litsentsiga riistvara tõttu on derivaadid ja kloonid täiesti legaalsed, kuid võltsingud on keelatud [6, 12, 13].

Hiinast tellides saab Arduinoga samaväärse funktsionaalsusega plaadi vaid mõne dollariga, kuid peab arvestama võimalusega, et toode on kehvema kvaliteediga kui originaal ning suurema ohuga, et jõuab kohale defektiga. Lisaks tuleb mõelda ka moraalsetele küsimustele, mis sõltuvad konkreetsest tootjast, kas firma panustab ka klienditoele ja tehnoloogiarendamisele ega tootmisel poleks kasutatud lapstööjõudu [13].

1.4 Erinevad Arduino tooted

Käesolevas punktis antakse ülevaade enim kasutatavatest Arduino arendus – ja laiendusplaatide mudelitest ning nende võimsusest, funktsionaalsusest ning kasutusvõimalustest. Tutvustatakse ka robotikaga alustamiseks kõige mugavamast võimalusest – stardikomplekti ning tuuakse välja võrdlus luksuslikuma originaalkomplekti ja odavama Arduino kloonil põhineva komplekti vahel.




1.4.1 Arduino arendusplaadid





Arduino arendusplaatidest on saadaval palju erinevaid mudeleid, mille abil on võimalik koostada erineva eesmärgi ja raskusastmega projekte. Mõned populaarsemad arendusplaatide mudelid ja nende info on toodud välja tabelis 1.


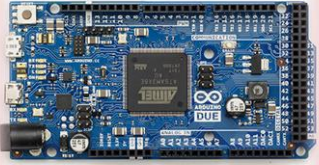


Arduinoga alustajatele on sobilikud odavama hinnaklassiga ning lihtsama disainiga UNO, 101, Pro (tabelis märgitud sinise taustaga) ja neile võimsuselt sarnased, kuid mõõduka väiksemad Mini, Micro ja Nano (tabelis märgitud lillaka taustaga). Keerukamate projektide jaoks on sobilikud Mega, Zero ja Due (tabelis märgitud punaka taustaga), mis võimaldavad suuremat taktisagedust, ühilduvust keerulisemate seadmetega, omavad rohkem välkmälu, mis on vajalik koodi hoidmiseks ning sisaldavad rohkem analoog- ja digitaalpesasid.

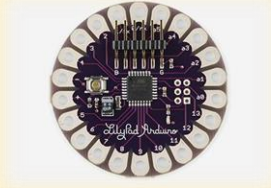
Ühendatud seadmetega seotud projektide (Internet of Things) jaoks on parim valik kahe protsessori, Wifi mooduli, USB-A pordi, micro SD kaardi pesa ja Linux'i operatsioonisüsteemiga Yún (tabelis märgitud rohelise taustaga). Tekstiilitöötlemise projektide jaoks on saadaval mõõtmelalt väikesed ja kaalult kerged Gemma ja erinevad versioonid enim moekunstis kasutatavatest riiete sisse õmmeldavast arendusplaadist LilyPad (tabelis märgitud kollase taustaga) [3].

Tabel 1. Arduino populaarsemad arendusplaadid [11, 14, 15, 16, 17, 18, 19, 20, 21, 22] [23, 24, 25, 26, 27, 28, 29]

Arendusplaat	Mikrokontrolleri mudel	Üldandmed	Joonis
Arduino UNO	ATmega328P	Tööpinge: 5V Soovitav sisendpinge: 7-12V Lubatud sisendpinge: 6-20V Taktisagedus: 16MHz Välkmälu: 32 KB	 <p>Joonis 3. Arduino UNO</p>
Arduino 101	Intel Curie	Tööpinge: 3.3V Soovitav sisendpinge: 7-12V Lubatud sisendpinge: 7-20V Taktisagedus: 32MHz Välkmälu: 196 kB	 <p>Joonis 4. Arduino 101</p>
Arduino Pro	ATmega168 või ATmega328	Tööpinge: 3.3V Lubatud sisendpinge: 3.35 -12V Taktisagedus: 8 MHz Välkmälu: 16 KB või Tööpinge: 5 V Lubatud sisendpinge: 5 - 12 V Taktisagedus: 16 MHz Välkmälu: 32KB	 <p>Joonis 5. Arduino Pro</p>

<p>Arduino Pro Mini</p>	<p>ATmega328</p>	<p>Tööpinge: 3.3V Lubatud sisendpinge: 3.35 -12 V Taktisagedus: 8 MHz Välkmälu: 32 kB või Tööpinge: 5V Lubatud sisendpinge: 5 - 12 V Taktisagedus: 16 MHz Välkmälu: 32 kB</p>	 <p>Joonis 6. Arduino Pro Mini</p>
<p>Arduino Micro</p>	<p>ATmega32U4</p>	<p>Tööpinge: 5V Soovitatav sisendpinge: 7-12V Lubatud sisendpinge: 6-20V Taktisagedus: 16MHz Välkmälu: 32 KB</p>	 <p>Joonis 7. Arduino Micro</p>
<p>Arduino Nano</p>	<p>Atmel ATmega168 või ATmega328</p>	<p>Tööpinge: 5V Soovitatav sisendpinge: 7-12V Lubatud sisendpinge: 6-20V Taktisagedus: 16MHz Välkmälu: 16 KB (ATmega168) või 32 KB (ATmega328)</p>	 <p>Joonis 8. Arduino Nano</p>
<p>Arduino Mega 2560</p>	<p>ATmega2560</p>	<p>Tööpinge: 5V Soovitatav sisendpinge: 7-12V Lubatud sisendpinge: 6-20V Taktisagedus: 16MHz Välkmälu: 256 KB</p>	 <p>Joonis 9. Arduino Mega 2560</p>

<p>Arduino Zero</p>	<p>ATSAMD21G18, 32-Bit ARM Cortex M0+</p>	<p>Tööpinge: 3.3V Soovitav sisendpinge: 7-12V Lubatud sisendpinge: 6-20V Taktisagedus: 48 MHz Välmälu: 256 KB</p>	 <p>Joonis 10. Arduino Zero</p>
<p>Arduino Due</p>	<p>AT91SAM3X8E</p>	<p>Tööpinge: 3.3V Soovitav sisendpinge: 7-12V Lubatud sisendpinge: 6-20V Taktisagedus: 84 MHz Välmälu: 512 KB</p>	 <p>Joonis 11. Arduino Due</p>
<p>Arduino Yún</p>	<p>ATmega32U4 ja Atheros AR9331 (Linux)</p>	<p>Tööpinge: 5V Lubatud sisendpinge: 5V Taktisagedus: 16 MHz Välmälu: 32 KB ja Tööpinge: 3.3V Taktisagedus: 16 MHz Välmälu: 16 MB Ethernet: IEEE 802.3 10/100Mbit/s WiFi: IEEE 802.11b/g/n</p>	 <p>Joonis 12. Arduino Yún.</p>
<p>Arduino Gemma</p>	<p>ATtiny85</p>	<p>Tööpinge: 5V Soovitav sisendpinge: 7-12V Lubatud sisendpinge: 6-20V Taktisagedus: 16MHz Välmälu: 8 kB</p>	 <p>Joonis 13. Arduino Gemma</p>

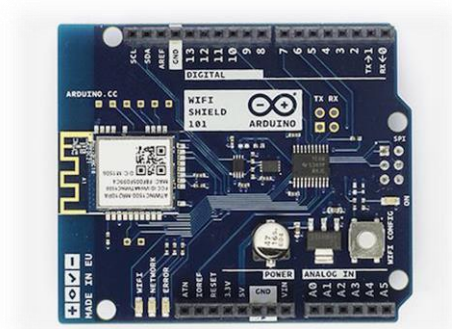
LilyPad Arduino Main Board	ATmega168 või ATmega328V	Tööpinge: 2.7-5.5 V Lubatud sisendpinge: 2.7-5.5V Taktisagedus: 8 MHz Välmälu: 16 KB	 <p>Joonis 14. LilyPad Arduino Main Board</p>
----------------------------------	-----------------------------	-----------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------

Huvitava projekti jaoks ei piisa enamasti ainult arendusplaadist, nii läheb ka Arduinoga töötamiseks enamasti vaja ka lisatarvikuid. Suuremate lisafunktsionaalsuste võimaldamiseks saab kasutada laiendusplaate.

1.4.2 Laiendusplaadid

Laiendusplaadid (i.k *shield*) ühilduvad arendusplaatidega ning võimaldavad keerukamate projektide jaoks lisafunktsionaalsust, nagu näiteks internetiga ühendamise võimalus, või Bluetoothi tugi [3].

Laiendusplaatide kasutamiseks on vaja arendusplaati. Enim kasutatavad laiendusplaadid on Motor Shield, mis on mõeldud lisamootorite (nt. samm-mootor) ühendamise lihtsustamiseks [30], Ethernet Shield kaabliga internetiühenduse võimaldamiseks [31], Proto Shield, mis on abiks skeemide disainimisel ning annab juurde lisapesasid suuremate projektide jaoks [32], GSM Shield, mis koos GPRS toega SIM kaardiga annab Arduinole mobiiltelefoni funktsionaalsused [33] ja WiFi Shield 101 (Joonis 15), mis võimaldab juhtmevaba internetiühendust [34].

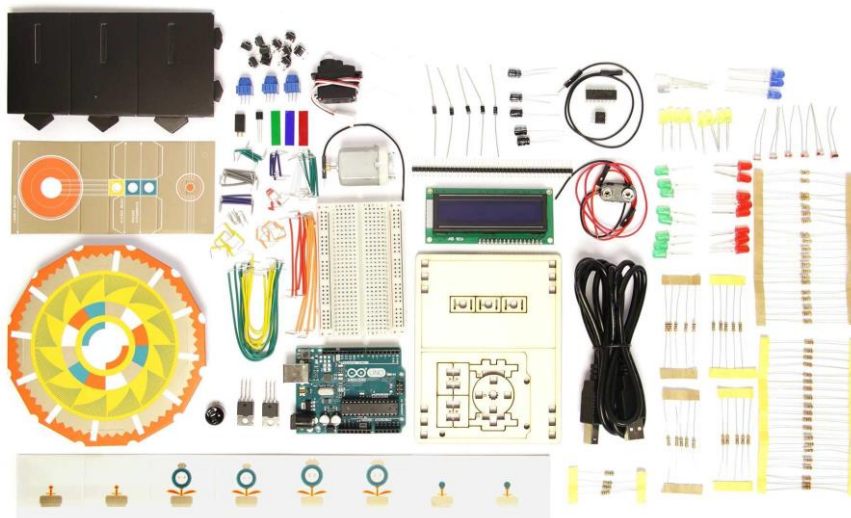


Joonis 15. Arduino WiFi Shield 101 [34]

Roboti ehitamiseks ega huvitavate katsete tegemiseks ei piisa ainult plaatidest, lisaks tulevad veel andurid, erinevad LED tuled ja kõik nende ühendamiseks vajalik. Seepärast on kõige parem alustada komplektiga, mis lisaks arendus- ja protoplaadile sisaldab ka vajalikke juhtmeid, maketeerimislauda, erineva värviga valgusdioode, takisteid ja nii alaloog kui ka digitaalandureid, mis tagavad hulgaliselt võimalusi erinevate projektide koostamiseks.

1.4.3 Arduinol põhinevad komplektid

Arduino ametlikud komplektid *Arduino Starter kit* ja *Arduino Basic Kit* (Joonis 16) on sisuliselt samad, ainuke erinevus on õpikus, *Starter kit* sisaldab 150-leheküljelist paberkandjal õpikut, *Basic kiti* ostjad saavad juurdepääsu sama sisuga e-materjalidele. Mõlemad sisaldavad UNO arendusplaati, üle 150 lisakomponendi, mille hulka kuuluvad erinevad andurid, LED-id, erineva takistusega takistid, nupud, juhtmed ja maketeerimislaud ning materjale 15 erineva projekti koostamiseks [35, 36].



Joonis 16. Arduino ametlik komplekt (Basic Kit) [36]

Originaalkomplektis sisalduvad juhendid on põhjalikud ning kõik komponendid kvaliteetsed ning luksuslikult pakendatud, mis teeb komplekti avamisest omaette elamuse [37].

Lisaks originaalkomplektidele on saadaval ka Arduino derivaatide ja kloonide komplekte, mille peamised komponendid sarnanevad originaalkomplektile, kuid mis sisaldavad

odavama hinna eest rohkem erinevaid andureid ja lisakomponente. Sellised komplektid sobivad küll õpiesmärkide täitmiseks, kuid neis sisalduvad tooted pole üldiselt nii kvaliteetsed ja süstematiseeritult pakendatud. Peamised erinevused komplektide vahel on välja toodud tabelis 2.

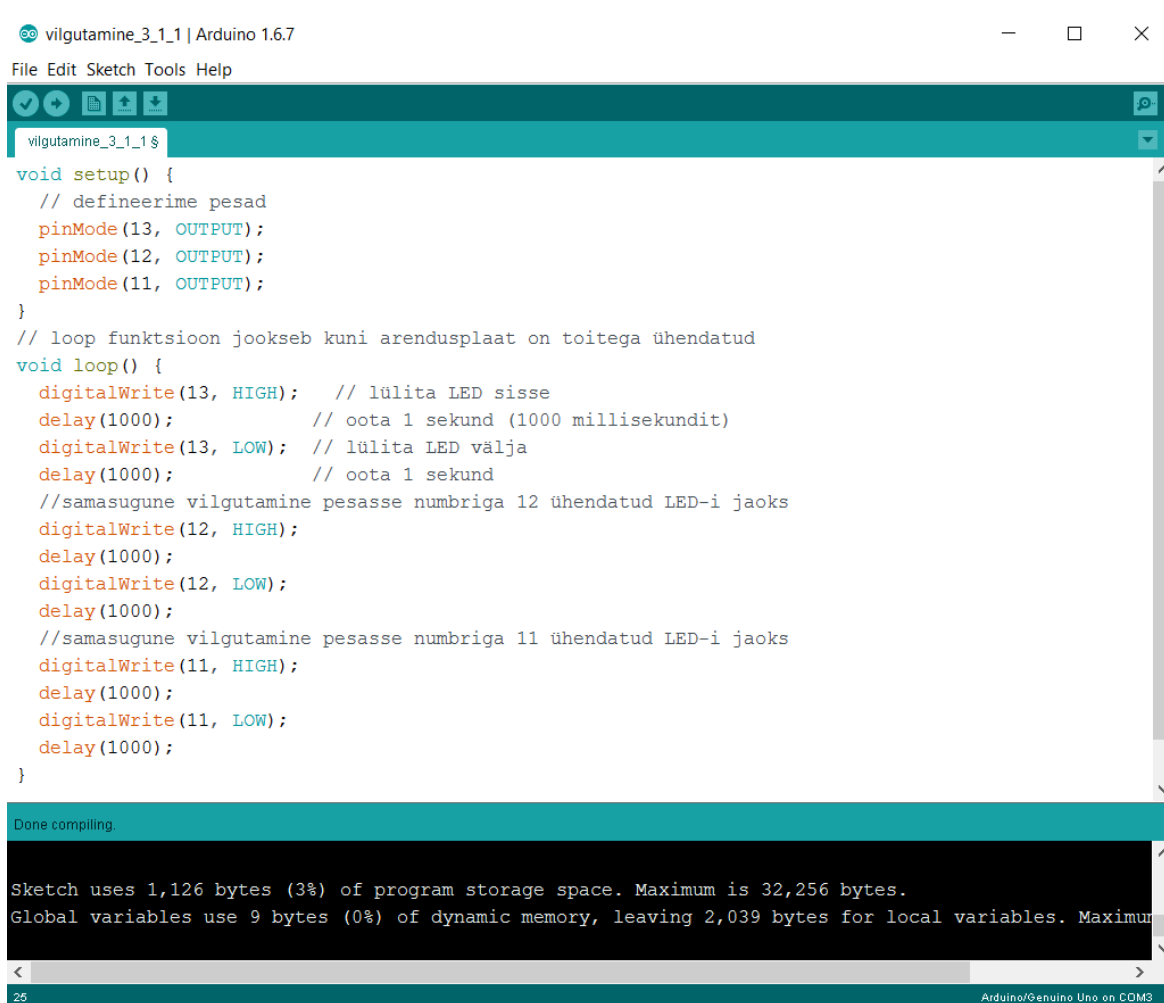
Tabel 2. Arduinol põhinevate komplektide võrdlus [38, 39]

	Originaalkomplekt Arduino Basic kit	Klooni komplekt UNO R3 Starter Kit for Infidduino
Hind seisuga mai 2016 (ostulink)	73.64 EUR / \$84 (Autodesk, Inc.)	46.05 EUR / \$52.53 (ebay)
Aredusplaat	Arduino UNO board rev 3	Infidduino Uno R3 Mainboard
Lülitite arv	10 + 1 kaldlüli	5 + 1 kaldlüli
LED-ide arv	28 (5 eri värvi) + 1 RGB	30 (3 eri värvi) + 1 RGB
Takistite arv	60 (7 erineva tekistusega)	60 (3 erineva takistusega)
Maketeerimislaud	30- realine	60 - realine
Juhtmete arv	72	65 + 3 soontega kaablit
Mootorid	1 x servo + 1 x DC	1 x samm + 1 x servo
Ekraan	1602 LCD	1602 LCD
Andurite arv	8 (2 erinevat)	8 (kõik erinevad)
Laiendusplaat	-	Protoplaat koos väikse maketeermislauaga
Teiste lisavidinate arv	24 (10 erinevat)	16 (13 erinevat)

Võrdlusest tulenevalt sisaldab Infidduino stardikomplekt kõige madalama hinna eest enim erineva funktsionaalsusega andureid ja lisatarvikuid, seega kasutame seda oma edasises töös. Käesoleva bakalaureusetöö teises peatükis kirjeldatakse täpsemalt antud stardikomplekti sisu ja kolmandas peatükis tuakse välja erineva raskusastmega näidisprojekte, mis kasutavad komplektis olevaid komponente.

1.5 Arduino programmeerimine

Arduinot ja sellel põhinevaid teiste tootjate plaate on võimalik programmeerida kõigis keeltes, mida on võimalik kompileerida masinkoodi [40], kuid lihtsam on kasutada mõnda arenduskeskkonda. Arenduskeskkondi on erinevatele sihtrühmadele. Mõned kogenumad programmeerijad eelistavad AVR Studiot selle keerukama ja rohkemate võimalustega veahalduse poolest [41], kuid riistvaralähedast programmeerimist alustavatele inimestele on siiski kõige parem valik joonisel 17 kujutatud Arduino integreeritud arenduskeskkond koos Arduino programmeerimiskeelega. Selle eelisteks on aktiivne kasutajatugi, lihtne ülesehitus, põhjalikud kasutajajuhendid ning kasutatavus kõigi levinumate operatsioonisüsteemidega [42].



```
vilgutamine_3_1_1 | Arduino 1.6.7
File Edit Sketch Tools Help
vilgutamine_3_1_1 $
void setup() {
  // defineerime pesad
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
}
// loop funktsioon jookseb kuni arendusplaat on toitega ühendatud
void loop() {
  digitalWrite(13, HIGH); // lülita LED sisse
  delay(1000);           // oota 1 sekund (1000 millisekundit)
  digitalWrite(13, LOW); // lülita LED välja
  delay(1000);           // oota 1 sekund
  //samasugune vilgutamine pesasse numbriga 12 ühendatud LED-i jaoks
  digitalWrite(12, HIGH);
  delay(1000);
  digitalWrite(12, LOW);
  delay(1000);
  //samasugune vilgutamine pesasse numbriga 11 ühendatud LED-i jaoks
  digitalWrite(11, HIGH);
  delay(1000);
  digitalWrite(11, LOW);
  delay(1000);
}
Done compiling.
Sketch uses 1,126 bytes (3%) of program storage space. Maximum is 32,256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2,039 bytes for local variables. Maximum
25 Arduino/Genuino Uno on COM3
```

Joonis 17. Ekraanitõmmis Arduino arenduskeskkonnast.

Antud keskkond ise on kirjutatud Javas ning inspireeritud keskkonnast Processing ja teistest vabavaralistest rakendustest. Programmeerimiskeel põhineb raamistikul Wiring ning on

väga sarnane C ja C++ keeltega [40]. Arenduskeskkonna kasutamise eeliseks on koodi värvimine, sulgude ühendamine, samuti toimuvad nii kompileerimine kui ka koodi plaadile laadimine ühe hiireklõpsuga. Kood kirjutatakse kahte funktsiooni: *setup()*, kuhu läheb osa koodist, mida käivitatakse vaid korra - vajalik algseadete määramiseks. Teise funktsiooni *loop()* kirjutatakse peamine kood, mida kutsutakse korduvalt välja nii kaua kuni plaat on sisse lülitatud [42]. Täpsema info arenduskeskkonna allalaadimise ja kasutamise kohta võib leida käesoleva töö peatükist 2.2.

1.6 Võrdlus Raspberry Pi-ga

Lisaks Arduinole kasutatakse koolirobootikas ka teisi platvorme, millel kõigil on oma eelised ja kitsaskohad, enamused neist on märgatavalt kallimad (hind umbes 100 eurot või üle). Seega on mõistlik võrrelda Arduinot sama hinnaklassi kõige populaarsema robootikaseadmega, milleks on Raspberry Pi (Joonis 18).



Joonis 18. Raspberry Pi Model 3 Model B [43]

Raspberry Pi on sisuliselt iseseisev arvuti, Arduino seevastu arendusplaat, plaadi hind sõltub mudelist ja lisadest. Ainult plaadi saab Arduino UNO puhul kätte odavamalt kui Raspberry Pi 3. Mõlemale platvormile on saadaval lisaandureid, Arduinole ka laiendusplaat. Huvitavamate projektide jaoks vajab Arduino siiski rohkem lisasid, millest tingituna on ka lisasid sisaldava komplekti hind veidi kallim. Näitajate järgi, mis on täpsemalt välja toodud tabelis 2, on Raspberry Pi 3 kiirema taktisagedusega ja omab 512 000 korda rohkem mälu kui Arduino UNO. Lisaks sisaldab Raspberry Pi ka operatsioonisüsteemi Linux ja kaabliga internetiühenduse võimalust, mis Arduinol võimalik vaid laiendusplaadiga [44]. Uuematel Raspberry Pi mudelitel on enamasti ka HDMI pesa, mis annab rohkem võimalusi teiste

seadmetega ühendamiseks. Samas ei nõua Arduino stabiilset sisendpinget ning analoogpesade olemasolu teeb analoogandurite (näiteks heliandur) ühendamise tunduvalt lihtsamaks.

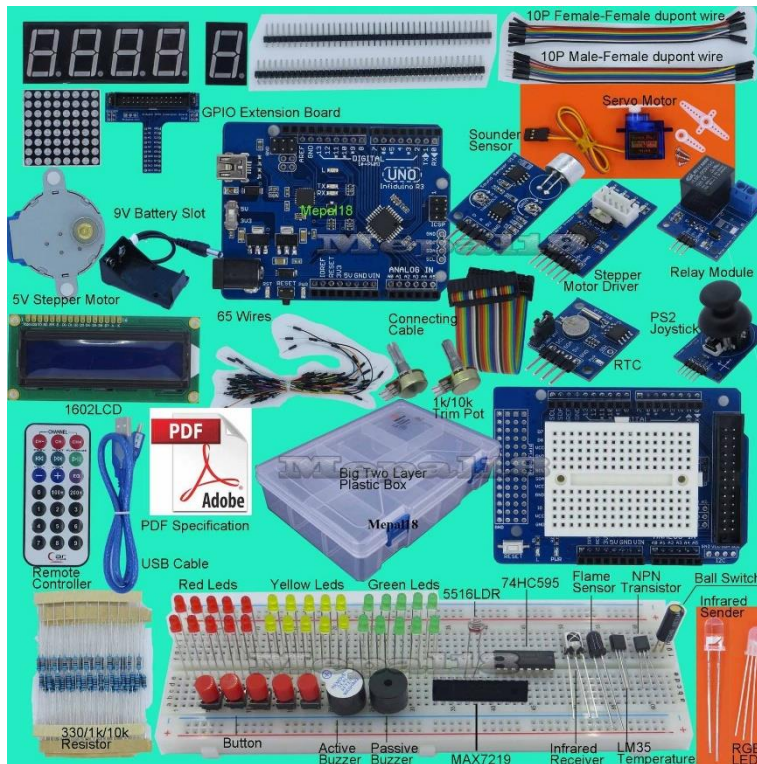
Tabel 3. Raspberry Pi Model 3 ja Arduino võrdlus [11, 43, 44]

	Arduino UNO R3	Raspberry Pi 3 Model B
Mälu	0,002 MB	1 GB (lisaks SD kaardi võimalus)
Taktisagedus	16 Mz	1.2GHz
Lubatud sisendpinge	6- 20V (soovitatav 7 - 12 V)	5 V
Ühendusvõimalused	14 digitaalpesa (millest 6 toetavad PWM väljundit) 6 analoogpesa	4 USB pesa 40 GPIO pesa 1 HDMI pesa 1 Ethernet pesa

Võrdlusest võib järeldada, et Raspberry Pi on sobilikum valik tarkvaraprojektide ja tarkvara testimise jaoks, kuid sellega on võimalik teha ka riistvara lähedasi lahendusi nagu näiteks nutirobot [45]. Seevastu annab Arduino lihtsus ja huvitavate andurite rohkus sellele eelise riistvaraprojektide puhul, kuna võimaldab kergema vaevaga lahendada ülesandeid, mis vajavad väliskeskkonnale reageerimist. Paljude erineva funktsionaalsusega, kuid ehituselt lihtsate andurite tõttu on Arduino ka ideaalseks alguspunktiks riistvaralähedase programmeerimise õppimiseks.

2 Stardikomplekti sisu ja seadistamine

Käesolevas peatükis tutvustatakse ühes Arduino kloni stardikomplektis (Joonis 19) sisalduvaid tarvikuid ning nende kasutusvõimalusi, näidatakse kuidas lisasid arendusplaadiga ühendada, arenduskeskkonda kasutada ning koodi arendusplaadile laadida.



Joonis 19. UNO R3 Starter Kit for Arduino/Infiduno [39]

2.1 Stardikomplekti sisu

Stardikomplekti kõige olulisem komponent on juba eelmisest peatükist tuttav UNO R3 mikrokontroller, sinna peale laaditakse soovitud kood ning külge ühendatakse kõik teised projektis kasutatavad komponendid. Erinevate tootjate komplektid võivad veidi varieeruda, antud töös on näidisprojektide tegemiseks kasutatud ebayst ostmishetkel 46.05 eurot maksnud Infiduno Ultimate UNO R3 stardikomplekti, kus on mõistliku hinna eest esindatud võimalikult paljud erinevad andurid [39]. Ülevaate selle komplekti tarvikutest annab tabel 4.

Tabel 4. Arduino stardikomplekti komponendid [39]

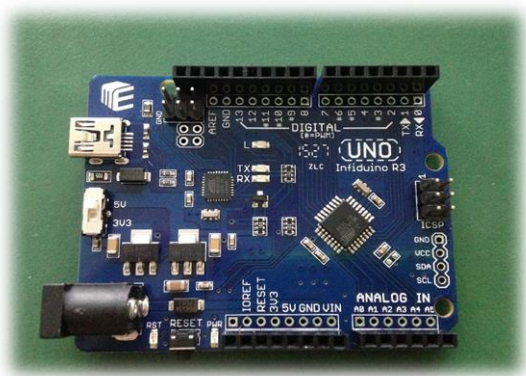
Nimetus	<i>ingliseelne nimetus</i>	Kogus komplektis	Viited selgitustele ja kasutustele käesolevas töös
Infiduinno Uno R3 arendusplaat	<i>Infiduinno Uno R3 Mainboard</i>	1	2.1.1, kõik 3. pt punktid
USB kaabel	<i>USB cable</i>	1	2.1.1, kõik 3. pt punktid
Juhtmed	<i>Wires</i>	2 x 10 soonega kaabel/ 65 x üksik	2.1.1, kõik 3. pt punktid
Maketeerimislaud	<i>Breadboard</i>	1 x suur ja 1x väike	2.1.1, 2.4, kõik 3. pt punktid
Takistid	<i>Resistors</i>	20 x 330 oom/ 20x 10Koom /20 x 1 Koom	2.1.3, kõik 3. pt punktid (va 3.8)
Tavalised 3 mm valgusdiodid	<i>3mm Red LEDs</i>	10 x punane/ 10 x roheline / 10 x kollane	2.1.2, 3.1, 3.2, 3.6
5 mm RGB valgusdiod	<i>5mm RGB LED</i>	1	2.1.2, 3.3, 3.4, 3.7
Nupulülitid /Kaldüliti	<i>Buttons with Cap / Ball switch</i>	5 x nupuga / 1 x kaldüliti	2.1.4, 3.2, 3.4
Potentsiomeeter (muuttakisti)	<i>Potentiometer</i>	1 x 1K/ 1 x 10K	2.1.9, 3.5
prototüüpimise laiendusplaat + GPIO laiendusplaat + 26-soonega ühenduskaabel	<i>Infiduinno Extension board + GPIO Extension Board + Connecting Cable</i>	1	2.4
Juhtkang	<i>Joystick</i>	1	2.1.5, 3.8
Pult	<i>Infrared Remote controller</i>	1	2.1.5, 3.9
Infrapuna vastuvõtja	<i>Infrared Receiver</i>	1	2.1.5, 3.9

Infrapuna saatja	<i>Infrared Sender</i>	1	2.1.9
1602A LDC ekraan	<i>1602A LCD Screen</i>	1	2.1.7, 3.5
Heliandur	<i>Sound Sensor</i>	1	2.1.8, 3.6
Fototakisti (valgusandur)	<i>Photoresistor</i>	1	2.1.8, 3.7
Samm-mootor + draiverplaat	<i>Stepper Motor + Driver Board</i>	1	2.1.6, 3.8
SG90 Servomootor	<i>SG90 Servo motor</i>	1	2.1.6, 3.8
Sumistid	<i>Buzzers</i>	1 x aktiivne / 1x passiivne	2.1.9
Leegiandur	<i>Flame Sensor</i>	1	2.1.9
Jalgade komplektid	<i>Pin headers</i>	1 x küljega 40 pesa/ 1 x sirge 40 pesa	2.1.7
MAX7219 kiip	<i>MAX7219 Chip</i>	1	2.1.9
74HC595 8-bitine nihkeregister	<i>74HC595 8bit Shift Register</i>	1	2.1.9
LM35 temperatuuriandur	<i>LM35 Temperature Sensor</i>	1	2.1.8, 3.10
SS8050 NPN bipolaartransistor	<i>SS8050 NPN Transistor</i>	1	2.1.9
7-segmendiline näidik	<i>Seven-segment Display</i>	1 x 1-numbriline / 1 x 4- numbriline	2.1.7, 3.9, 3.10
8 * 8 LED maatriks	<i>8*8 Dot Matrix Led</i>	1	2.1.7
Kellaaja moodul	<i>RTC (Real Time Clock) module</i>	1	2.1.8
Relee moodul	<i>Relay Module (5V- 10A)</i>	1	2.1.9
Patareihooldja (9 V patarei jaoks)	<i>9 V Battery Holder</i>	1	2.1.9

Täpsemad kirjeldused tabelis kujutatud toodete kohta on toodud järgnsvates alampeatükkides.

2.1.1 Kõikides projektides kasutatavad komponendid

Mikrokontrollerit (Joonis 20) võib vaadata kui projekti südant, selle külge ühendatakse kõik ülejäänud projektis kasutatavad komponendid ning mälus hoitakse ülesannet täitvat koodi.



Joonis 20. Infidino UNO mikrokontroller

USB kaabel on vajalik mikrokontrolleri ühendamiseks arvutiga nii toite eesmärgina kui ka koodi plaadile laadimise jaoks. Maketeerimislauale ühendatakse juhtmete abil andurid ja indikaatorid, selle olemasolu teeb võimalikuks komponentide ühendamise ilma jootmiseta, mistõttu on see hea lahendus prototüüpimiseks ja koolirobootika projektide lahendamiseks.

2.1.2 Valgusdiodid

Valgusdiodid on pooljuhid, mis suudavad muuta elektrienergia valguseks. Pikemat jalga, millele rakendatakse positiivne pinge nimetatakse katoodiks, lühemat jalga, millele rakendatakse negatiivne pinge, anoodiks [46]. Tavalised kahe jalga valgusdiodid (Joonis 21) põlevad ainult ühte värvi.



Joonis 21. Eri värvi tavalised valgusdiodid

Ühel LED-il eri värvide kuvamiseks on välja töötatud 4-jalaga RGB valgusdiod (Joonis 22), kus üks jalg on kas anood või katood (sõltuvalt valgusdiodi tüübist) ja ülejäänud kolm tähistavad punast, rohelist ja sinist LED-i .



Joonis 22. RGB valgusdiod

Selline ehitus teeb võimalikuks ühe LED-i sees värvide vahetamise. Valgusdioode kasutatakse nii vooluringi ülesannetes vooluringi osana kui ka digitaalväljundina valgusmängu loomiseks ja erinevate andurite testimiseks.

2.1.3 Takistid

Takistid on vajalikud olukordades, kus tahame piirata komponendini jõudvat voolutugevust nagu näiteks valgusdioode kasutavates projektides, kuna liiga tugev elektrivool kahjustab LED-i, lühendades selle eluiga märgatavalt. Samas liiga tugeva takisti kasutamisel on valgusdiodid tuhmid ning projekt vähem efektne. Erineva takistusega takisteid on võimalik eristada värvitriipude järgi, värvikoodide lugemiseks on kõige lihtsam kasutada mõnda internetis leiduvat kalkulaatorit, näiteks G-Sound OÜ kodulehel asuv takistite värvikoodide kalulaator ([link](#)). Komplektis on takistid pakendatud 20 kaupa ning takistus on pakendiribale.



Joonis 23. Pakendis takistid

Sobiva takisti valimiseks peame välja arvutama, kui suurt takistust vaja on, seda võib teha käsitsi valemiga

$$R = \frac{(\text{Arduino toide } (5 V) - \text{valgusdiodi päripinge (keskmiselt } 2 V))}{\text{valgusdiodi nimivool } (0.02 A)}$$

või kasutada mõnda kalkulaatorit, näiteks linear.org asuv LED kalkulaator ([link](#)) [47]. Saame vastuseks (sõltuvalt valgusdiodist) midagi 150Ω kanti, see on minimaalne väärtus, millega meie takisti tohib olla. Soovitav on võtta umbes 20 Ω võrra tugevam. Kuna täpselt sobiva väärtusega takistit ilmselt pole, siis tuleb otsustada komplektist kõige lähema, suurema takistusega takisti kasuks.

2.1.4 Lülitid

Lülitid on kasutusel vooluringi kontrollimiseks. Nupuga lülitite (Joonis 24) puhul ühendab lüliti vajutamise vooluringi. Kaldlüliti sõltub ühendus lüliti asendist, vooluring on ühendatud, kui kaldlüliti on püstises asendis ehk lüliti sees olev kuulike puutub ühendustega kokku.



Joonis 24. Tavalised nupuga lülitid

Lüliteid on lisaks vooluringi mehaanilisele osale võimalik ka vaadata kui andureid ja programmeerida digitaalsisenditena. Sellisel juhul tuleb kasutada takistit.

2.1.5 Projekti juhtimiseks vajalikud tarvikud

Projektide üle suurema kontrolli saavutamiseks kuuluvad komplekti ka pöidlaga juhitud juhtkang (Joonis 25) ja koos infrapuna vastuvõtjaga kasutatav pult (Joonis 26).



Joonis 25. Juhtkang

Juhtkangi saab programmeerida vastavalt kangi asendile soovitud tegevust sooritama. Arduino Leonardo arendusplaat võimaldas ka juhtkangi kasutamist arvutihiirena, kuid selles töös kasutatav UNO R3 seda võimalust ei paku. Seega on antud komplekti puhul juhtkangi parim kasutusala erinevate mootorite liigutamine.



Joonis 26. Komplektis sisalduv pult

Parim viis projektide eemalt kontrollimiseks on puldi ja infrapuna vastuvõtja (Joonis 27) abil.



Joonis 27. Infrapuna vastuvõtja

Selleks tuleb tuvastada puldi nuppude signaalid ning seada need vastavusse soovitud tegevustega. Selliste ülesannete jaoks võib kasutada nii komplektis sisalduvat pulti või ka vabalt mõnda muud infrapuna saatjaga (pulti otsas olev väike läbipaistev andur) toodet, näiteks tavalist televiisoripulti. Erinevaid pulte kasutades tuleb arvestada, et programm tuleb kohandada kasutatava puldi signaalidele.

2.1.6 Mootorid

Servomootor on tagasisidega mootor, mida saab signaali abil soovitud asendisse juhtida. Signaali katkemisel naaseb mootor tasakaaluasendisse. Turul on saadaval eri pöördekraadiga servomootoreid. Poolpöörde servomootorid (Joonis 28) võimaldavad 180-kraadist liikumist ning nende kõige levinum kasutusala on robotite jäsemete liigutamine. Täispöörde ehk 360-kraadiseid mootoreid saab kasutada robotite rataste veomootoritena [48].



Joonis 28. 180-kraadine ehk poolpöörde seromootor

Käesolevas komplektis sisaldus erineva pikkusega mootori külge ühendatavaid tiivikuid ning üks poolpöörde servomootor, mida saab juhtkangi abil näiteks lüliti kustutama juhtida.

Samm-mootor on mootor, mis pöörleb täisringe sammude kaupa. Erinevalt servost puudub samm-mootoril tasakaaluasukoht, kuid kuna võimaldavad määrata kiirust ning distantsi, siis on kindlaks määratud asukohta liikumine märgatavalt täpsem. Samm-mootoreid on nii unipolaarseid (Joonis 29) kui ka bipolaarseid. Erinevus seisneb mähiste ühendustes ja juhtimises, unipolaarset mootorit on kergem juhtida, kuna piisab võimalusest voolu välja lülitada, bipolaarse mootori juhtimiseks tuleb aga muuta polaarsust. Unipolaarset mootorit saab ühendada kui bipolaarset, kuid vastupidine ühendamine pole võimalik.



Joonis 29. Samm-mootor koos draiveriga

Samm-mootori kasutamiseks tuleb ühendada ka draiver, mis määrab sammu impulsid, liikumissuuna ning sammualgoritmi. Sobivad kasutamiseks kohtades, kus vaja täpsust, näiteks tindiprinterites tindi pea liigutamiseks. Samas ei ole samm-mootor parim valik roboti veomootorina, kuna võrreldes servomootoriga on nad väiksema erivõimsuse, suurema voolutarbega ning raskemini juhitud [49].

2.1.7 Erkaanid ja näidikud

1602A LDC (Joonis 30) on kahest reast ja 16 veerust koosnev vedelkristallekraan, info, näiteks andurite abil saadud mõõtmistulemuste, kuvamiseks. Iga andmekuvamiskast on lahutusvõimega 2 x 7 pikslit. Ekraani kontrollimiseks saab kasutada Arduino arenduskeskkonnaga kaasas olevat teeki *LiquidCrystal library*. Ekraani saab kontrollida nii 4-biti kui ka 8 -biti režiimis. Ainult teksti kuvamiseks piisab 4 -bitist.



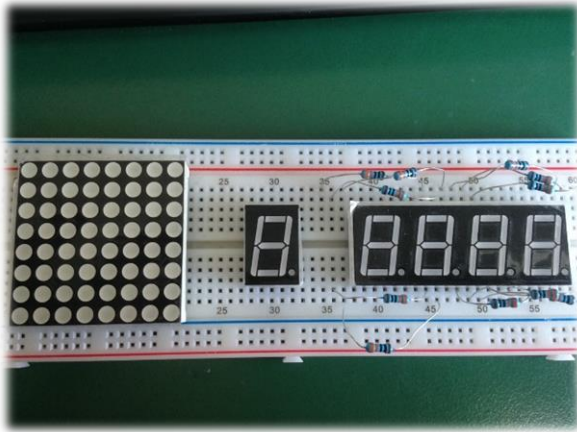
Joonis 30. 1602A LDC ekraan

Joonisel 30 kujutatud ekraani saab kontrollida 16 pesa abil, millest esimene, VSS tähistusega, on maandus, järgmisse, VDD märgiga ühendatakse toide (5 V), kolmas, V0, on kasutusel kontrasti muutmiseks, neljas (RS) tähistab registrivalikut andme- ja juhisteregistri vahel. Andmeregistrist tulev info kuvatakse ekraanil, juhisteregistri järgi valitakse järgmine tegevus. Viies, RW märgistusega, pesa määrab kas informatsiooni loetakse või kirjutatakse, mis ainult kirjutamiseks ühendatakse lihtsalt maandusega. Kuues (E), reguleerib, kas registritesse kirjutamine on lubatud või mitte. Järgmised kaheksa pesa, tähistusega D0-D7, on andmepesad, mille seisundid (*HIGH* või *LOW*) tähistavad registrisse kirjutatavaid ning sealt loetavaid bittide. Kaks viimast pesa on LED taustvalguse anood (A) ja katood (K) [50].

Paljudes komplektides, kaasaarvatud käesolevas, on ekraan transpordi lihtsustamise eesmärgil pakendatud jalgadest eraldi. Sellisel juhul tuleb enne kasutama hakkamist jalad

pesade külge joota. Järgneva peatüki näidisprojektides kasutatud ekraani külge on joodetud komplektis sisalduvad sirged jalad.

LCD ekraani kasutamine pole komplekti ainuke info kuvamise võimalus. Numbrilisi näitajaid saab hästi edasi anda ka 7-segmendilistel LED valgusega näidikutel. Samal tehnoloogial põhineb ka 8x8 LED maatriks.



Joonis 31. 8x8 LED maatriks, 1- ja 4- numbrilised 7-segmendilised näidikud

Ühest numbrist koosneva info, näiteks puldil vajutatud numbrilise kuvamiseks sobib ka 1-numbriline näidik, kuid mõõtmisandmete nagu temperatuur või kellaeg näitamiseks tuleb kasutada 4-numbrilist displeid. Maatriksil on võimalik kuvada näiteks üksikuid tähti või 64-punkti abil eristatavaid pildikesi.

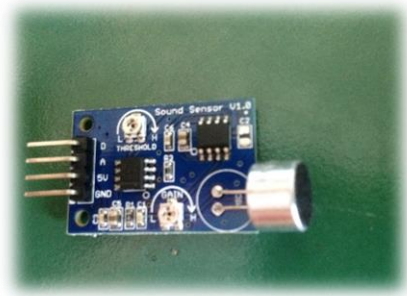
2.1.8 Keskkonnale reageerivad andurid

Fototakisti (Joonis 32) on valgustundlik andur, mida kasutatakse valguse olemasolu ja intensiivsuse määramiseks. Mida pimedam on, seda suurem on takistus (kuni 1 megaoom), valguse peale takistus väheneb sõltuvalt valguse intensiivsusest. Ereda valguse puhul võib takistus langeda mõne oomini. Fototakisteid kasutatakse näiteks kaamerate valgusandurina, valgusvihi järgi liikuvates robotites ning tänavalampide valgustundlikuks muutmiseks [51].



Joonis 32. Fototakisti [52]

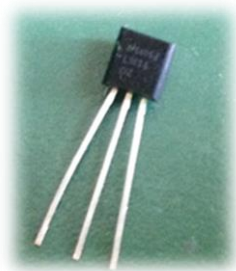
Heliandur (Joonis 33) on mikrofoni ja heliandmete töötusplaadi kombineerimisel saadud andur, mis suudab mõõta müratugevust.



Joonis 33. Heliandur

Komplektis olevat andurit on võimalik kasutada nii analoog kui ka digitaalandurina. Digitaalsisendina kasutamine on mõttekas, kui soovime infot ainult müra olemasolu kohta. Olukordades, kus soovime saada andmeid ka müratugevuse kohta, tuleb ühendada andur analoogpesaga [53].

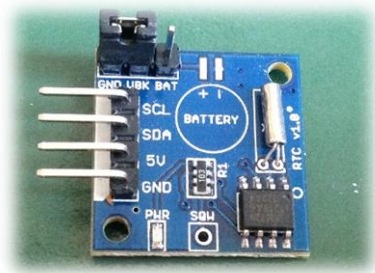
Temperatuuriandur (Joonis 34) on analoogandur, mis mõõdab ümbritseva keskkonna temperatuuri. Saadud vastuse saame kerge vaevaga teisendada endale sobivaks ühikuks (näiteks kraadideks Celsiuse järgi) [54].



Joonis 34. Temperatuuriandur LM35

Temperatuuriandurit kasutatakse tavaliselt õhutemperatuuri mõõtmiseks, kuid võib ka proovida, kui adekvaatset tagasisidet annab kehatemperatuuri mõõtmine näiteks peopesast.

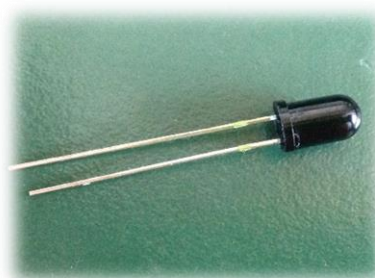
Kellaaja moodul (Joonis 35) ehk RTC (Real Time Clock) on vajalik praeguse ajahetke tuvastamiseks. Seda saab kasutada digitaalse kella loomisel või ajaarvestuse pidamiseks hetkedel, kus ülejäänud süsteem on toitest eemaldatud [55].



Joonis 35. Kellaaja moodul

Toitest sõltumatu ajaarvestuse pidamiseks tuleb RTC moodulil olevale patareikohale lisada patarei.

Leegiandur (Joonis 36) suudab tuvastada infrapuna lained lainepikkusega 760 nm ~ 1100 nm. Tänu sellele on see tundlik leegist tulevate kiirgusele, kuid nähtav valgus komponendi ei mõjuta [56].



Joonis 36. Leegiandur

Andur reageerib elavale tulele, mille tõttu koolides kasutamine ei ole praktiline. Samas on sel komponendil oma roll targa maja projektides ohutust tagava andurina.

2.1.9 Teised komponendid

Infrapuna saatja (Joonis 37) on seade, mille abil on võimaik saata infrapuna vastuvõtjale loetavat signaali elektroonikaseadmete juhtimiseks.



Joonis 37. Infrapuna saatja

Selle abil on näiteks võimalik ehitada televiisoripult või edastada signaale erinevate arendusplaatide vahel.

Potentsiomeeter on muuttakisti, mille abil on võimalik reguleerida näiteks valgusdioodide eredust või helitugevust. Välimuselt sarnaneb potentsiomeeter kruvile ning seda on saadaval erinevate takistustega [57].



Joonis 38. Potentsiomeetrid

Käesolevas komplektis sisaldub kaks potentsiomeetrit, 10-kilo-oomine ja 1-kilo-oomine.

MAX7219 kiip (Joonis 39) on näidiku draiver, mis sobib kasutamiseks ühise katoodiga displeide puhul.



Joonis 39. MAX7219 kiip

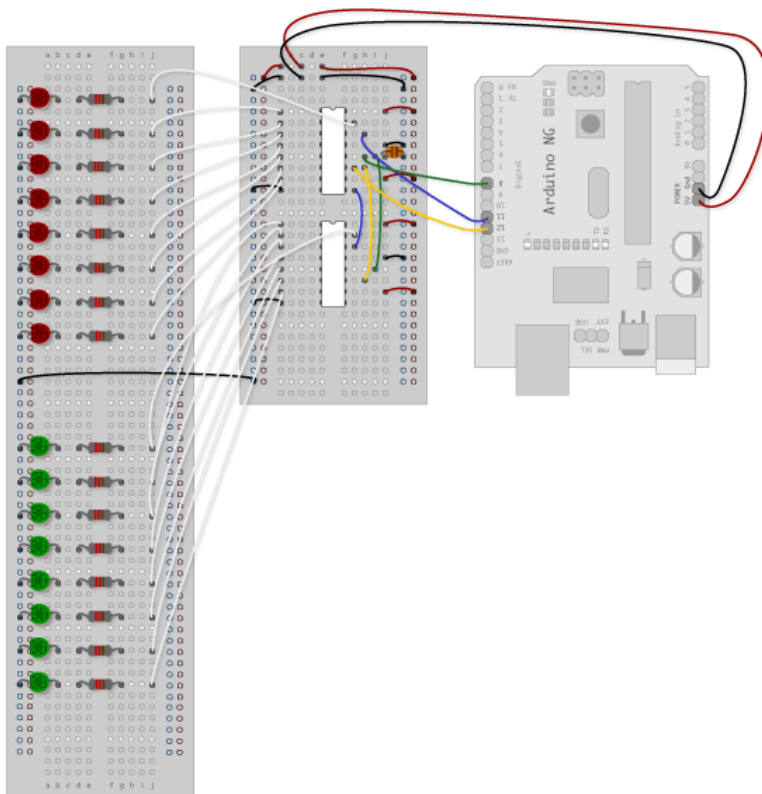
Selle abil on võimalik lihtsustada nii mitme 7-segmendiliste näidiku üheaegset ühendamist kui ka 8 x 8 maatriksite kasutamist [58].

Nihkeregister on vajalik olukordades, kus arendusplaadil olevatest pesadest jääb soovitud ülesande täitmiseks väheks [59]. Komplektis sisalduv 74HC595 8-bitine nihkeregister (Joonis 40) võimaldab kaheksa digitaalpesa asemel kasutada korraga kolme.



Joonis 40. 74HC595 8-bitine nihkeregister

Digitaalpesade puudus või esineda näiteks siis, kui soovime põneva valgusmängu jaoks ühendada korraga palju valgusdioode, veel võimsama tulemuse saame mitme nihkeregistri üheaegsel kasutamisel (Joonis 41).



Joonis 41. Näide kahe nihkeregistri üheaegsest kasutamisest [59]

Nagu näha jooniselt 41, siis saame paralleelselt kasutada mitut nihkeregistrit nii, et arendusplaadil oleks endiselt hõivatud vaid kolm digitaalpesa [59].

SS8050 NPN (Joonis 42) on bipolaartransistor ning võimaldab ühendada Arduinoga kõrgemate elektriliste nõuetega seadmeid.

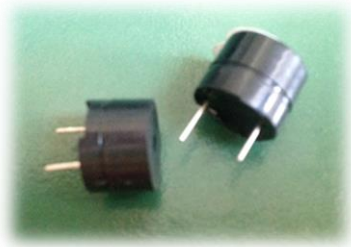


Joonis 42. SS8050 NPN transistor

NPN on transistori tüüp, mille kasutamisel ühendatakse kollektor mootoriga ja emitter maandusega. Ilma transistorita ei ole digitaalpesadega võimalik ühendada komponente,

mille elektrilised nõudmised on kõrgemad kui arendusplaadi poolt pakutav 40mA vool ja 5V pinge. Kõige enam on transistoritest kasu mootorite ühendamisel, kuna suuremate robotite ehitamisel kasutatavatel võimsamatel mootoritel on kõrged tehnilised nõudmised [60]. Käesoleva transistori abil on võimalik ühendada komponente, mis nõuavad kuni 1,5A voolu ja 40V pinget [61].

Sumistid on helisignaali tekitavad komponendid, mida saab kasutada näiteks äratuskellades, uksekellades või hoiatussignaali tekitamiseks ohutusandurites.



Joonis 43. Passiivne ja aktiivne sumisti

Komplektis sisaldub kaks erinevat sumistit – aktiivne, mis suudab ise signaali toota ning passiivne, millele tuleb Arduinoga soovitud signaal ette anda.

Relee moodul on elektriline lüliti, mille abil on võimalik kontrollida Arduinost tunduvalt kõrgema pingega töötavaid elektroonikaseadmeid. Relee kontrollimine Arduino toimub digitalWrite funktsiooni abil, antakse ette käsk sisse (HIGH) või välja (LOW) lülitamiseks [62]. Projektis sisalduva relee abil on võimalik näiteks 9 V patarei kontrollimine. Komplektis sisaldub patareialus, kuid patarei tuleb ise juurde osta.

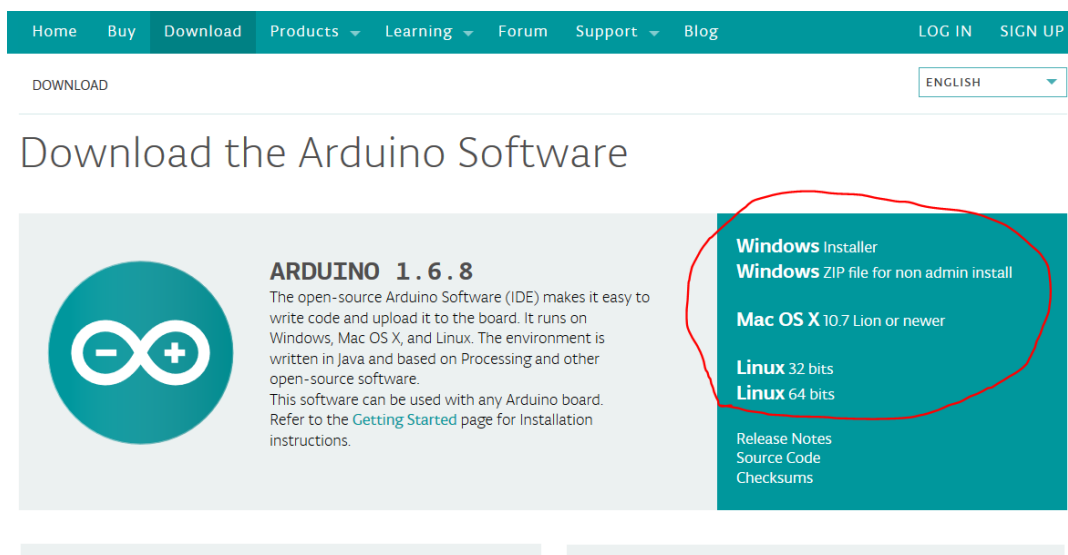


Joonis 44. Relee moodul ja patareihoidja

Releed on tähtsal kohal targa maja projektides erinevate elektroonikaseadmete lülitamiseks, näiteks kohvimasina käivitamine äratuskella helina peale.

2.2 Arenduskeskkonna paigaldamine ja arendusplaadi ühendamine

Arduino kasutamiseks on kõigepealt vaja alla laadida koodi kirjutamiseks ja kompileerimiseks vajalik tarkvara, näiteks Arduino arenduskeskkond. Keskkonna leiab Arduino ametlikult kodulehet joonisel 45 kujutatud “*Download*” vaatest ([link](#)) ja selle allalaadimine on tasuta.



Joonis 45. Arenduskeskkonna alla tõmbamise esimene samm

Allalaadimise alustamiseks tuleb valida joonisel 45 punasega märgistatud menüüst kasutatavale operatsioonisüsteemile sobiv versioon. Windowsi kasutajatel on valida installeri, mis nõuab käivitamiseks administraatoriõigusi ja kokkupakitud programmi vahel. Kuna enamikes kooli ja koduarvutites on kasutusel Windows, siis antud peatükis näitame, kuidas installeerida Arduino arenduskeskkonda Windowsile.

Pärast sobiva versiooni valimist avanevas vaates (Joonis 46) küsitakse, kas soovite Arduinole annetada, kui hetkel seda teha ei soovi, siis valige joonisel 46 punasega märgistatud “*JUST DOWNLOAD*”.

Support the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)

SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **7,010,570** TIMES. IMPRESSIVE! THIS IDE IS NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS. HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING IT TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEIT. YOU CAN HELP ACCELERATE THE DEVELOPMENT OF THE ARDUINO IDE BY CONTRIBUTING TOWARDS THE EFFORT OF MAKING IT BETTER.

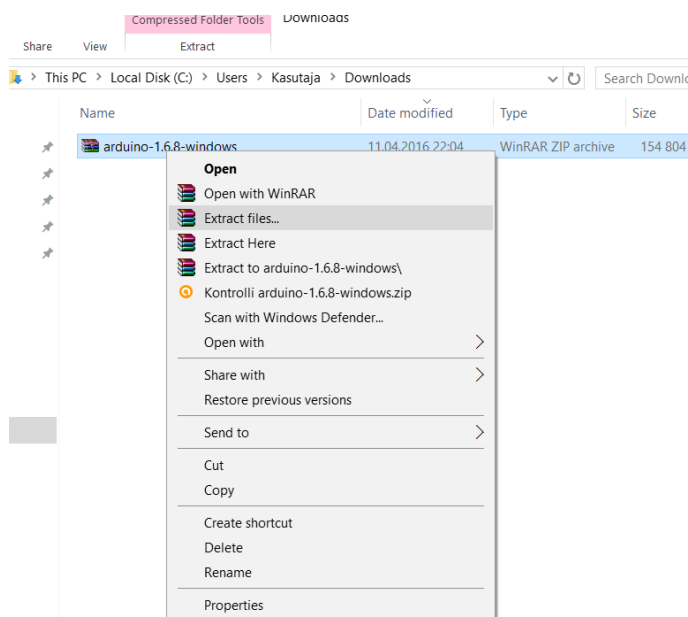
\$3 \$5 \$10 \$25 \$50 OTHER

JUST DOWNLOAD CONTRIBUTE & DOWNLOAD

Joonis 46. Arenduskeskkonna alla tõmbamise samm 2

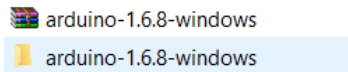
Kui allalaadimine on lõppenud, siis Windowsi installeri puhul topeltklõps installeri exe formaadis failile, nõustuge tingimustega, valige programmile meelepärane asukoht ja vajutage “*Install*”. Pärast edukat installeerimist tuleb kiri “*Completed*”, seejärel klõpsake ainukest aktiivset nuppu “*Close*”. Installeerimine on edukalt lõppenud, arenduskeskkonda on võimalik avada töölauale ilmunud otseteest ja ka kaustas, kuhu programmi salvestasite, asuvast arduino.exe failist.

Kokkupakitud versiooni puhul avage kaust, kuhu zip faili salvestasite, klõpsake failile parema hiireklahviga ning valige “*Extract files...*” (Joonis 47).



Joonis 47. ZIP faili lahti pakkimine

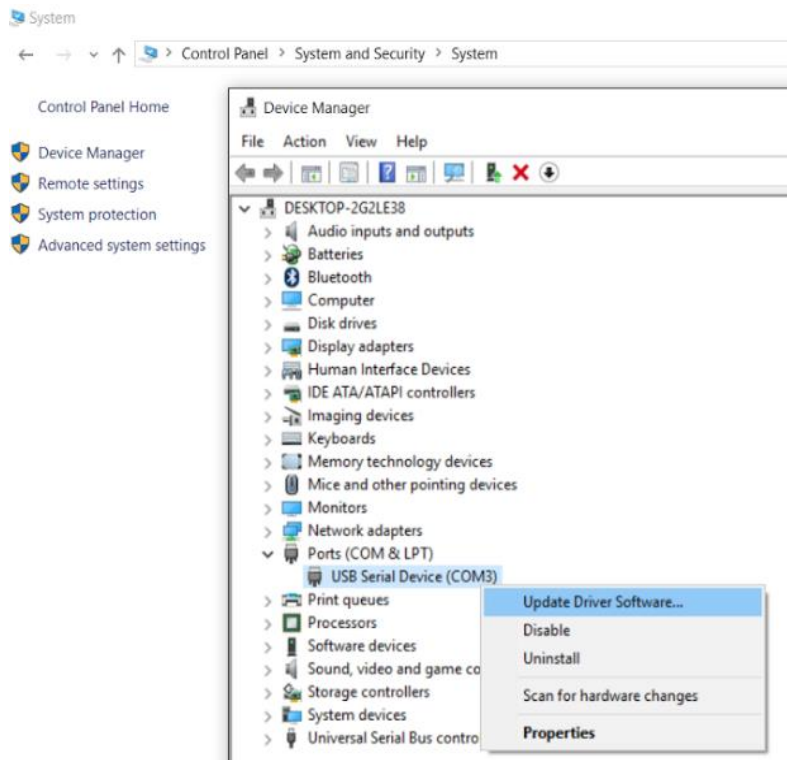
Seejärel klõpsake “Ok” ning samasse kausta ilmub zip failiga sama nimega kaust (Joonis 48), kus sees on soovitud tarkvara. Arenduskeskkonna käivitamiseks topeltklõps kaustas olevale `arduino.exe` failile.



Joonis 48. Lahtipaikitud kaust arenduskeskkonnaga

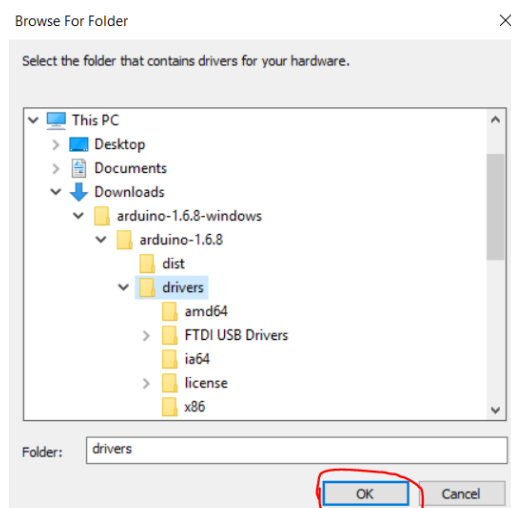
Nüüd kui vajalik tarkvara on olemas, võib ühendada arendusplaadi arvutiga. Selleks ühendada eelmises alampeatükis tutvustatud USB kaabli väiksem ots mikrokontrolleri külge ja teine arvutiga. Ühenduse saamisel süttib mikrokontrolleril roheline PWR märgistusega tuld. Esmakordsel ühendamisel käivitatakse automaatselt draiverite installatsioon (alates Windows XP kuni Windows 10 puhul), oodata kuni see lõpetab.

Kui mingil põhjusel draiverite installatsioon esmaühendamisel ei õnnestu, siis avada arvutis “Control Panel” (“Juhtpaneel”) -> “System and Security” (“Süsteem ja turve”) -> “System” (“Süsteem”). Avenenud vaates avada “Device manager” (“Seadmehaldur”). Leida üles ripploendist “Ports (COM & LPT)” (“Pordid (COM & LPT)”) ja selle all peaks olema meie arendusplaat, mille nimeks antud näites on USB Serial Device COM3, kuid mõnda teist plaati kasutades võib nimi olla veidi erinev. Juhul kui “Ports (COM & LPT)” peaks ripploendist puuduma, siis tuleks vaadata “Other Devices” (“Teised seadmed”) alla ning seal peaks olema “Unknown Device” (“Tundmatu seade”). Seejärel tuleb klõpsata seadmed parempoolse hiireklahviga ning valida draiverite tarkvara uuendamine (Joonis 49).



Joonis 49. Draiverite tarkvara uuendamise

Avanenu aknas valida draiverite tarkvara sirvimine oma arvutist ("*Browse my computer for driver software*"). Nüüd tuleb otsida draiverite alamkaust kaustast, kuhu meil Arduino tarkvara on installeeritud, milleks tarkvara paigaldamiseks installerit kasutades on tõenäoliselt (C:\Program Files (x86)\Arduino\drivers). ZIP failina installatsiooni puhul tuleb navigeerida kausta, kuhu ZIP sai lahti pakitud ning valida kaust "drivers". Arduino draiverite kaust leitud, vajutada ok (mitte lasta end häirida sellest, et draiverite kaust sisaldab alamkaustu ja valida see mille nimi on lihtsalt "drivers") (Joonis 50).



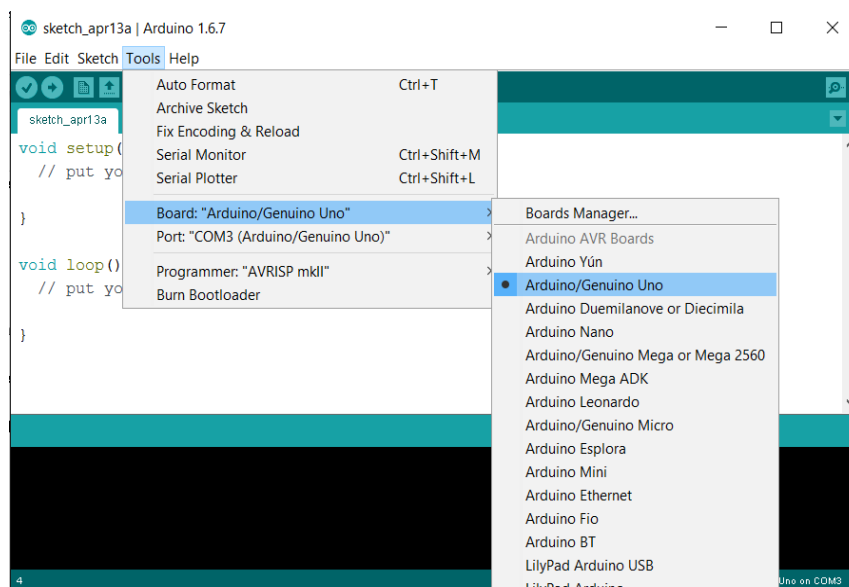
Joonis 50. Draiverite tarkvara leidmine enda arvutist

Nüüd tuleb vajutada lihtsalt “Next” ja oodata, et Windows lõpetaks draiverite installatsiooni.

Sellega on ettevalmistused tehtud, koodi testimisest, kirjutamisest ja plaadile laadimisest tuleb juttu järgmises alampeatükis.

2.3 Koodi kirjutamine ja plaadile laadimine

Koodi kirjutamiseks tuleb avada Arduino arenduskeskkond ning valida “Tools” alt millise arendusplaadiga tegu (Joonis 51), kuna käesolevas töös vaadeldavas stardikomplektis sisalduv plaat on võrdväärne Arduino UNO-ga, siis selle seaded sobivad.

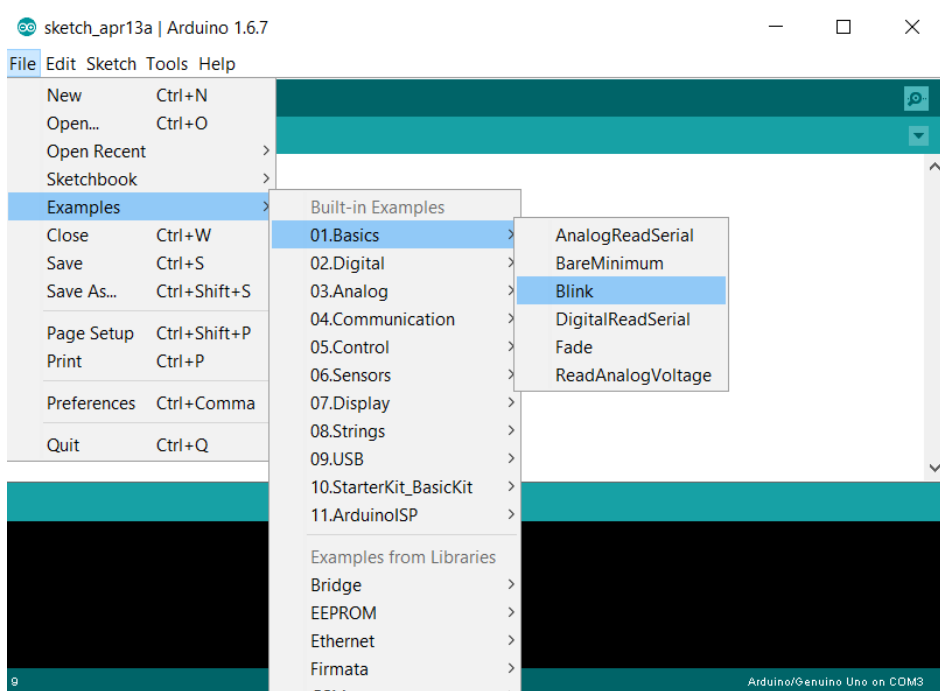


Joonis 51. Plaadi mudeli valimine

Järgmisena tuleb valida port, mida saab teha Joonisel 51 kujutatud “Port” valikust (asub “Board” valiku all). Kui valikus on ainult üks port, siis ongi juba õige port valitud, kui porte on mitu ja tekib kahtlus, kas valitud on õige, siis jätta meelde millised pordid valikus on, ühendada Arduino arvuti küljest lahti ja avada sama menüü uuesti, nüüd on üks port kadunud ja see puuduolev port ongi selle arendusplaadi oma. Seejärel saab plaadi tagasi ühendada, õige port on taas tagasi ja saame selle valida.

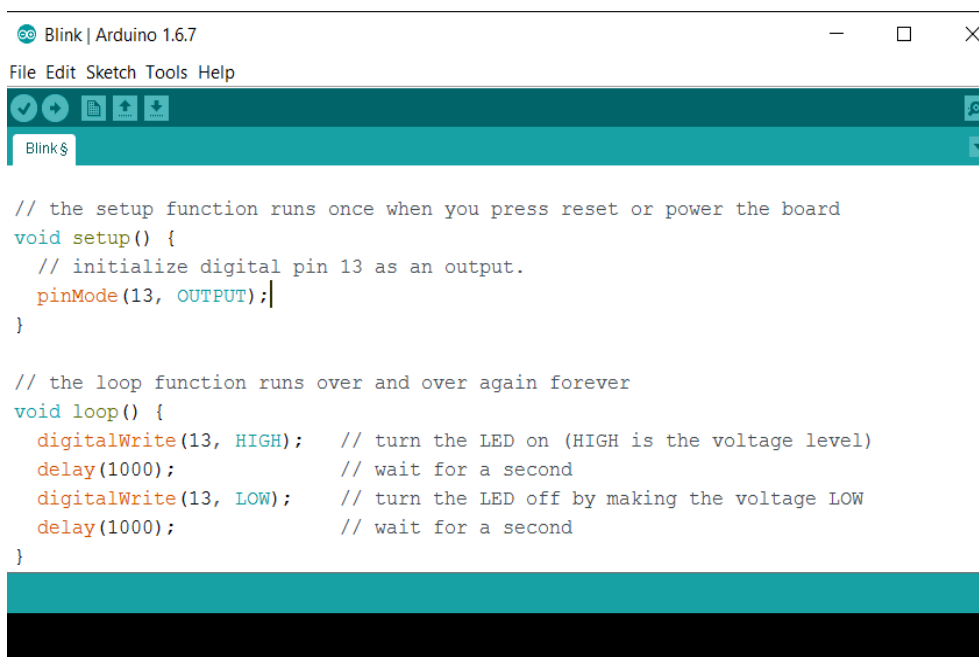
Esimeseks programmiks sobib väga hästi valgusdiodi vilgutamine, mis vilgutab plaadil olevat LED-i (tähistatud L tähega). Antud programm on juba arenduskeskkonnaga kaasas ning selle ja paljude teiste näitekoodide leidmiseks tuleb valida menüü “File” alt

“Examples”. Vilgutamiskood asub kaustas “01-Basics” ning selle nimi “Blink”, antud tee on kujutatud ka joonisel 52.



Joonis 52. Tee vilgutamisnäiteni

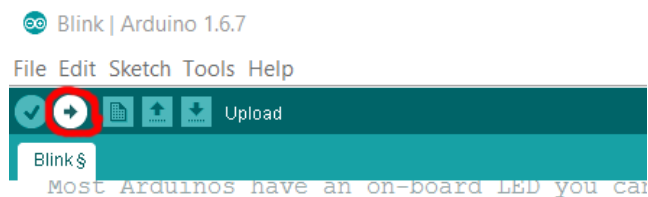
Seejärel avaneb uus arenduskeskkonna aken (Joonis 53), kus on lahti Vilgutamise projekti kood. Suur osa avanevas aknas olevast tekstist on kommentaarid, need read koodi toimimist ei mõjuta. Mitmerealine kommentaar paigutatakse märkide /* */ vahele ning üherealist kommentaari tähistab //.



Joonis 53. Vilgutamisnäite kood

Funktsioonis setup määratakse millisesse väljundiks digitaalpesa numbriga 13, kuna plaadil olev LED on ühendatud selle pesaga. Funktsioonis loop, mida täidetakse nii kaua kuni plaat on vooluallikaga ühendatud, lülitatakse LED sisse, oodatakse 1 sekund (1000 millisekundit), lülitatakse LED välja ja oodatakse uuesti 1 sekund.

Nüüd on teorias selge mida programm tegema peaks ning saame selle plaadile laadida. Soovi korral võib koodi ka enne lihtsalt kompileerida, vajutades rohelisel ribal olevale linnukese märgile (“*Verify*”), see harjumus tuleb hiljem ise kirjutatud koodide puhul kasuks, kuna aitab vigu avastada. Kompileerimiseks ja plaadile laadimiseks valida programmi üläääres asuvalt tegevusribalt noolemärk ehk “*Upload*” (Joonis 54).



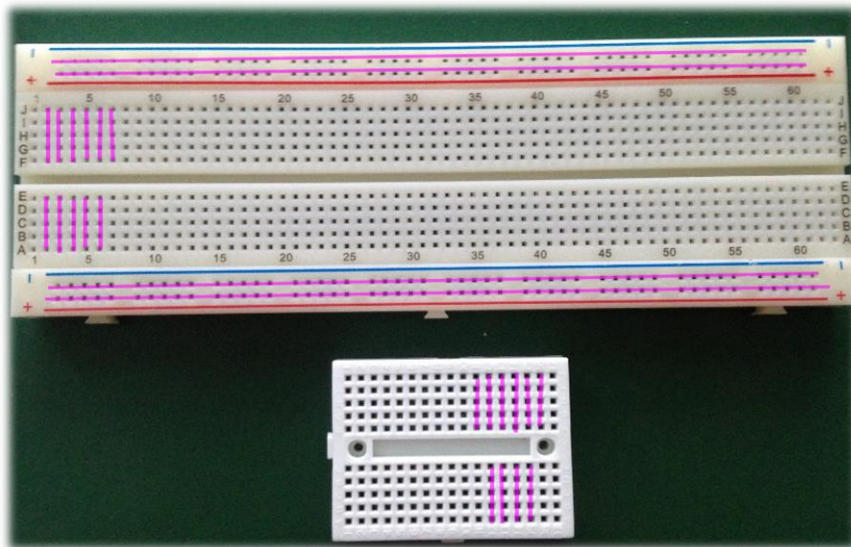
Joonis 54. Koodi plaadile laadimine

Kui kõik läks hästi, siis koodi all olevasse musta kasti tuleb valge kirjaga teade palju mälu kood võtab ning kasti kohal olevale rohelisele ribale ilmub väike kiri “*Done uploading*”. Vea ilmumisel läheb roheline riba oranžiks, selle ilmub veateade ning musta kasti kuvatakse punase kirjaga täpsem info vea kohta.

2.4 Maketeerimislaua ja protoplaadi kasutamine

Käesolevas töös on enamus projekte koostatud kasutades peatükis 2.1 tutvustatud maketeerimislauda, kuna see annab võimaluse mugavalt ja jootevabalt koostada erinevatest komponentidest koosnevaid vooluringe. Selline vajadus tekib valgusdiodide ühendamisel ning ka siis, kui arendusplaadiga tuleb ühendada rohkem kui üks andur.

Joonisel 55 on roosaga märgitud maketeerimislaua ühenduses olevad kontaktid.



Joonis 55. Maketeerimislaual ühenduses olevad kontaktid

Jooniselt 55 on näha, et vool liigub suure maketeermislaua äärmistes ridades (tähistatud + ja -) mööda rida. Pluss märgiga tähistatud reale ühendatakse toide (arendusplaadil tähistatud 5 V), miinus märgiga reale maandus (arendusplaadil tähistatud GND). Kõigis keskmistes ridades liigub vool mööda veergu, need read on mõeldud komponentide paigutamiseks.

Väiksemal protoplaadil (joonisel 55 alumine) liigub vool mööda veerge, mille tõttu tuleb maandus ise pisikeste juhtmejuppide abil üle plaadi vedada. Samas on võimalik kompaktsuse tõttu mõned juhtmed ka ära jätta. Näiteks saame ära jätta valgusdioode kasutavas projektis takisti ja digitaalväljundi vahelise juhtme ning ühendada takisti otse sobivasse digitaalpesasse. Valgusdioodide ühendamist seletava ja teiste lihtsamate projektidega on võimalik tutvuda käesoleva töö järgmises kolmanda peatüki esimestes projektides.

Keerukamate projektide puhul, kus maketeerimislaual hakkab ruumi väheks jääma, on võimalik protoplaati ja maketeerimislauda koos kasutada. Selle teeb lihtsamaks samuti peatükis 2.1 tutvustatud GPIO laiendusplaat.

3 Õppematerjalid

Käesolevas peatükis koostatakse juhendid, mille abil on võimalik lahendada nii Arduinoga alustavatele sobiliku raskusastmega projekte kui ka natuke keerulisemaid ülesandeid. Kõikides õppematerjalides on kasutatud ainult eelmises peatükis lähemalt tutvustatud UNO R3 Infiduinno stardikomplektis sisalduvaid komponente. Õppematerjalid koosnevad praktilistest näiteülesannetest, kus näidatakse, kuidas ühendada ja programmeerida erinevaid komplektis sisalduvaid andureid, LED tulesid, lüliteid ja teisi komponente. Materjalide läbimine annab ülevaate Arduino platvormi võimalustest, analoog ja digitaalandurite tööst, elektroonika ühendamisest ning põhilistest programmeerimises kasutatavatest võtetest. Joonised on koostatud vabavaralise prototüüpide koostamise programmiga fritzing.

3.1 Erinevad võimalused LED vilgutamiseks ja FOR-tükkel

Käesolevas alampeatükis ühendatakse LED- tulesid maketeermislauale, proovitakse läbi erinevad vilgutamise võimalused ja tutvustatakse programmeerimises palju kasutatavat FOR-tsükli.

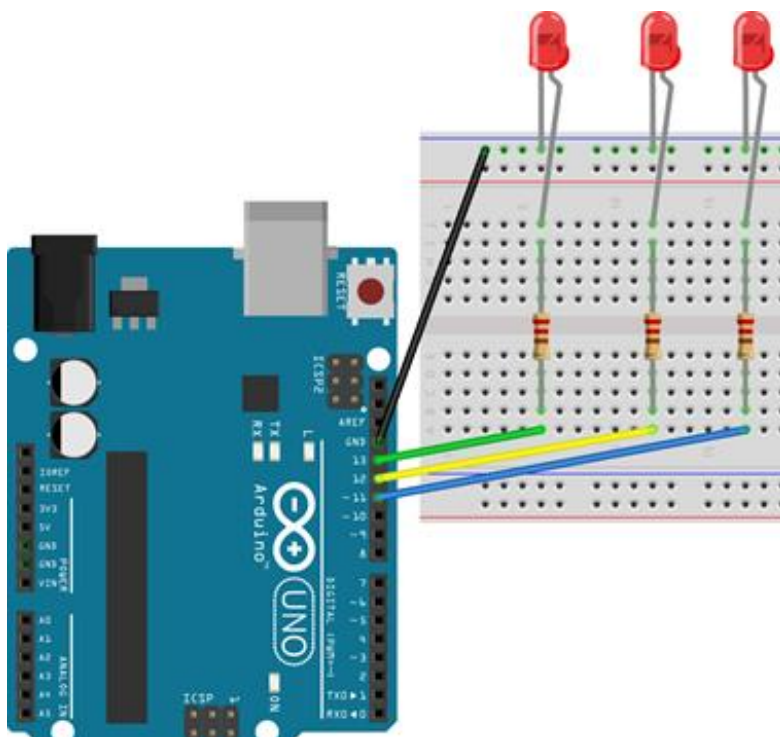
Projekti koostamiseks läheb vaja:

1. 1 x UNO mikrokontroller
2. 1 x USB kaabel
3. 1 x maketeerimislauad
4. 3 x vabalt valitud värvi LED-id
5. 3 x 330 Ω takistid
6. 4 x tavalised üksikud isane-isane juhtmed

Enne komponentide ühendamist tuleb veenduda, et arendusplaat oleks arvuti küljest (ja muudest toidetest) lahti ühendatud.

Ühendada kolm LED-i rööptühenduses maketeerimislauale nii, et katood (LED-i lühem jalg) oleks ühendatud maketeerimislaua miinusreale ning anood (LED-i pikem jalg) ühendatud maketeerimislaua keskel takistiga ning seejärel digitaalsignaali, antud näites on kasutusel digitaalpesad numbritega 13, 12 ja 11. Miinusreale tuleb lisada ka maandusjuhe (ühendatud

digitaalpesade kohal olevase GND pesasse). Ühendused võimalik üle kontrollida ka järgnevalt jooniselt 56.



Joonis 56. Kolme valgusdiodi ühendamise maketeerimislauale

Nüüd on aeg ühendada arendusplaat USB juhtme abil arvutiga ning avada arenduskeskkond. Kõigepealt võime uuesti proovida juba eelmisest peatükist tuttavat Blink näidet (*File -> Examples -> Basics -> Blink*). Laadime koodi kõigepealt muutmata kujul plaadile. Selle peale peaks vilkuma hakkama nii plaadil olev LED kui ka number 13 digitaalpesasse ühendatud valgusdiodid.

Kahel ülejäänud lambil sellise koodiga otstarve puudub. Proovime kopeerida Blink projektis olevat koodi nii, et ka pesadega 12 ja 11 ühendatud LED-id kasutust leiaksid, näide selle kohta on toodud järgnevas koodiplokis (koodi kommentaarid muudetud eestikeelseks).

```
void setup () {  
  // defineerime pesad  
  pinMode(13, OUTPUT);  
  pinMode(12, OUTPUT);  
  pinMode(11, OUTPUT);  
}  
// loop funktsioon jookseb kuni arendusplaat on toitega ühendatud
```

```

void loop() {
    digitalWrite(13, HIGH); // lülita LED sisse
    delay(1000);           // oota 1 sekund (1000 millisekundit)
    digitalWrite(13, LOW); // lülita LED välja
    delay(1000);           // oota 1 sekund
    //samasugune vilgutamine pesasse numbriga 12 ühendatud LED-i jaoks
    digitalWrite(12, HIGH);
    delay(1000);
    digitalWrite(12, LOW);
    delay(1000);
    //samasugune vilgutamine pesasse numbriga 11 ühendatud LED-i jaoks
    digitalWrite(11, HIGH);
    delay(1000);
    digitalWrite(11, LOW);
    delay(1000);
}

```

Joonis 57. Kood valgusdiiodide järjest vilgutamiseks

Saime koodi mis küll töötab, aga on ebaefektiivne, kuna sarnast funktsionaalsust kutsutakse mitu korda uuesti välja ning mida rohkem LED-e kasutusel on, seda tüütumaks selline olukord muutuks. Selle vältimiseks on mõistlik panna kirja sama asi FOR- tsükli abil sarnaselt järgnevas koodiplokis kujutatud näitele.

```

void setup() {
    // defineerime pesad
    pinMode(13, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(11, OUTPUT);
}
//loop funktsioon jookseb kuni arendusplaat on toitega ühendatud
void loop() {
    // tsükkel, mis käib läbi pesad 11 kuni 13, paneb LED-id põlema ja
    kustutab ära
    //suurendame pesa numbrit iga ringiga
    for (int pesa = 11; pesa <= 13; pesa++) {
        digitalWrite(pesa, HIGH); // lülita LED sisse
        delay(1000); // oota 1 sekund (1000 millisekundit)
        digitalWrite(pesa, LOW); // lülita LED välja
        delay(1000); // oota 1 sekund
    }
}

```

Joonis 58. Kood järjest vilgutamiseks kasutades FOR-tsükli

Pauside eemaldamiseks enne järgmise LED-i süttimist tuleb eelnevast koodist eemaldada viimane FOR - tsükli sees olev rida (*delay(1000);*). Kõikide LED-ide korraga põlema jätmiseks tuleb eemaldada FOR - tsükli kehast kolm viimast rida (*delay(1000)*, *digitalWrite(pesa, LOW);* ja *delay(1000);*). Kõigi lampide järjest põlema panemiseks ja kustutamiseks tuleb lisada kustutamise jaoks uus tsükkel, mida võib näha ka järgnevas koodiplokis.

```
void setup() {
  // defineerime pesad
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
}
//loop funktsioon jookseb kuni arendusplaat on toitega ühendatud
void loop() {
  // tsükkel, mis käib läbi pesad 11 kuni 13
  //pane LED-id põlema suurendame pesa numbrit iga ringiga
  for (int pesa = 11; pesa <= 13; pesa++) {
    digitalWrite(pesa, HIGH); // lülita LED sisse
    delay(1000); // oota 1 sekund (1000 millisekundit)
  }
  delay(1000); // oota veel üks sekund kui kõik LED-id põlevad
  //tsükkel kustutamiseks
  for (int pesa = 11; pesa <= 13; pesa++) {
    digitalWrite(pesa, LOW); // lülita LED välja
    delay(1000); // oota 1 sekund
  }
}
```

Joonis 59. Kood valgusdiiodide korraga süütamiseks ja kustutamiseks

Sellela oleme läbi proovinud mõned valgusdiiodide erinevatest vilgutamisvõimalustest ja tutvunud FOR- tsükliga. Lisaks võib proovida veel näiteks kõigi korraga süütamist ja kustutamist ning vilkumiskiiruse muutmist (*delay* funktsioonile vähem millisekundeid ette andes).

3.2 Lüliti ühendamise ja LEDi reageerimine vastavalt lüliti asendile vooluringis ja digitaalsignaali abil IF-lausega

Käesolevas alampeatükis näidatakse esmalt lihtsa vooluringi koostamist, mis sisaldab LED- tulesid ning erinevaid lüliteid. Seejärel lüliti abil programmi kontrollimist, mille käigus tutvustatakse ka programmeerimises levinud IF-tingimust.

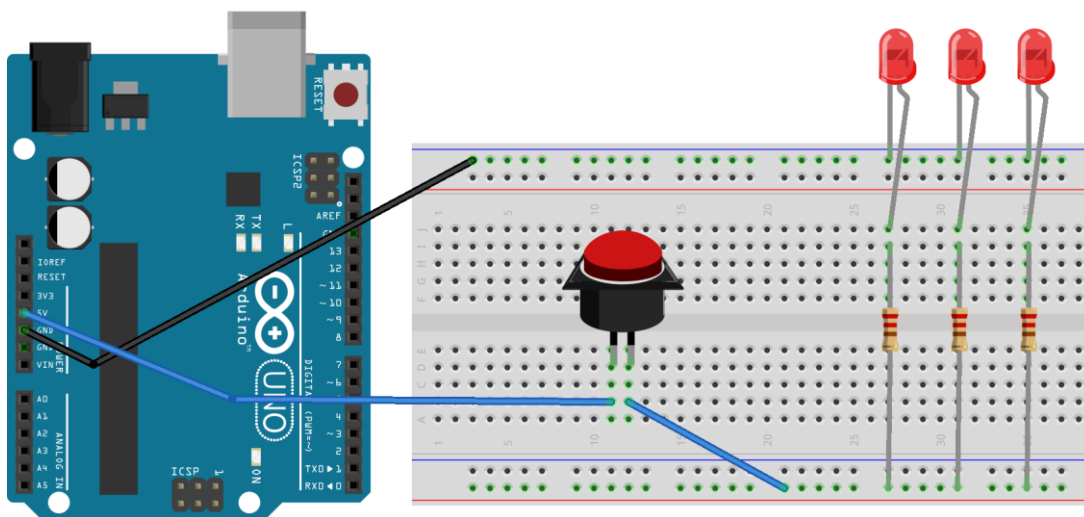
Projekti koostamiseks läheb vaja:

1. 1 x UNO mikrokontroller
2. 1 x USB kaabel
3. 1 x maketeerimislaud
4. 3 x vabalt valitud värvi LED-id
5. 3 x 330 Ω takistid
6. 1 x 10k Ω takisti
7. 4 x tavalised üksikud isane-isane juhtmed
8. 1 x tavaline lüliti
9. 1 x kaldlüliti

Enne komponentide ühendamist tuleb veenduda, et arendusplaat oleks arvuti küljest (ja muudest toidetest) lahti ühendatud.

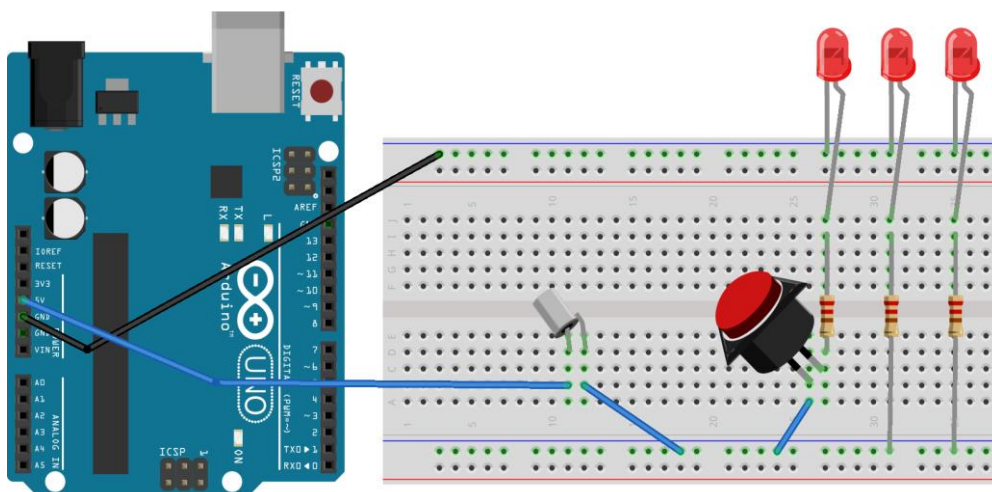
Algas sarnaneb peatükis 2.6.1. tehtud LED-ide vilgutamise projektile, tuleb ühendada kolm LED-i rööpühenduses maketeerimislauale nii, et katoode (LED-i lühem jalg) oleks ühendatud maketeerimislaual miinusreale ning anood (LED-i pikem jalg) ühendatud maketeerimislaual keskel takistiga, seekord digitaalsignaali ühendamist pole vaja, selle asemel ühendame takisti ühe jala hoopis maketeerimislaual ääres olevale + reale.

Seejärel paigutada lüliti mugavasse kohta maketeerimislaual keskel nii, et lüliti jalad oleks paigutatud erinevatesse veergudesse. Komponentid paigas, lisada maandusjuhe (GND) LED-ide katoodidega samale miinusreale ning toitejuhe (5 V) lüliti ühe jalaga samasse veergu. Vooluringi lõpetamiseks lisada juhe lüliti teise jala ja + rea vahele sarnaselt joonisel 60 kujutatule.



Joonis 60. Lüliti ja valgusdioodide ühendamine

Testimiseks ühendada arendusplaat toite eesmärgil arvutiga, kuna tegu on lihtsa vooluringiga, kus arendusplaat on vaid vooluallikaks, siis arenduskeskkonda ega koodi vaja ei lähe. Kui kõik ühendused toimivad, siis peaks nupule vajutades kõik kolm valgusdioodi põlema minema ning nupu lahti laskmisel uuesti kustuma. Kui ühe lülitiga katse töötab, siis võib proovida sisuliselt sama asja mitme lülitiga, lisades LED-idele individuaalseid lüliteid. Joonisel 61 on kasutatud peamise lülitina kaldlülitit ning ühele LED-ile on lisatud tavalülitit.

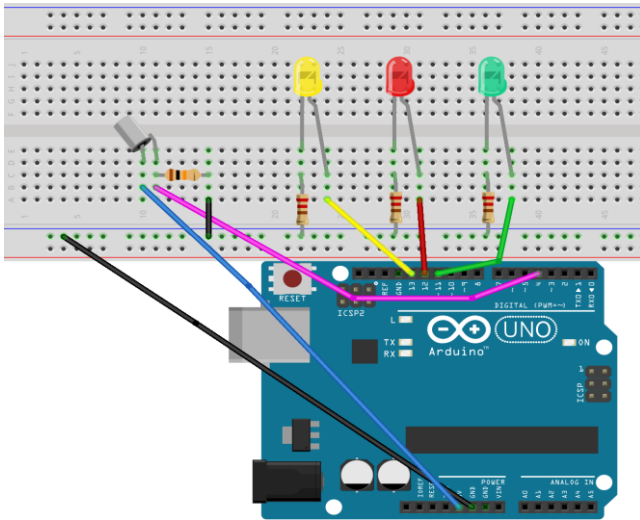


Joonis 61. Nupulülitit, kaldlülitit ja valgusdioodide ühendamine

Nüüd peaksid tavaasendis põlema kaks LED-i ja kolmas siis kui lüliti on alla vajutatud. Maketeerimislaual kallutamisel peaksid kustuma kõik tuled.

Kui tavaline vooluring selge, siis võib proovida olukorda, kus lülitiga kontrollitakse signaali, näiteks lülitada vilkuvaid LED-e. Selleks ühendada arendusplaat aruti küljest lahti ning paigutada LED-id maketeerimislaual natuke ümber, nii et katood (lühem jalg) oleks

ühendatud läbi takisti maandusega ning anood (pikem jalg) digitaalsignaali, antud näites on LED-id ühendatud pesadesse 11, 12, ja 13. Lüliti (pole oluline, kas kasutada kaldlüliti või tavalist) toitejuhe võib jääda endisele kohale, kuid LED-ide kontrollimiseks tuleb lisada ühendus digitaalsignaali. Selleks lisada vahele maandusega ühendatud 10 kilo-oomine takisti ja sarnaselt joonisel 62 kujutatule võib digitaalsignaali ühendada näiteks pesasse numbriga 4.



Joonis 62. Lüliti ühendamise digitaalsisendina

Digitaalsignaali kontrollimiseks lüliti abil ühendada arendusplaat arvutiga ning laadida arenduskeskkonna abil plaadile järgnev kood, kus kontrollitakse IF-tingimuse abil, kas lüliti on sisse lülitatud, kui tingimus kehtib täidetakse eelmisest alampeatükist tuttav FOR-tükkel. Tingimuse mittekehtimise korral kustutatakse kõik valgusdiodid.

```
//lülitamise algväärtus
int on_lylitatud = 0;
void setup() {
  // defineerime LED-ide pesad
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
  // defineerime lüliti pesa sisendina
  pinMode(4, INPUT);
}
//loop funktsioon jookseb kuni arendusplaat on toitega ühendatud
void loop() {
  //loeme digitaalsignaali kas lüliti on vajutatud
  on_lylitatud = digitalRead(4);
  //kontrollime, kas signaal on olemas
  if (on_lylitatud == HIGH){
    //kui on, siis kordame eelmise alampeatüki vilgutamisülesannet
    //ainult, et natuke kiirema vilkumisega
    for (int pesa = 11; pesa <= 13; pesa++) {
```

```

digitalWrite(pesa, HIGH); // lülita LED sisse
delay(100); // (100 millisekundit)
digitalWrite(pesa, LOW); // lülita LED välja
}
}
else{
//signaali puudumise korral, tuled kustu
digitalWrite(13, LOW); //LED välja
digitalWrite(12, LOW); // LED välja
digitalWrite(11, LOW); // LED välja
}
}
}

```

Joonis 63. Kood lüliti kasutamiseks digitaalsisendina kõikide LED-ide korraga kontrollimiseks

Nüüd peaksid kaldlüliti puhul tavaasendis kõik LED-id vilkuma ning maketeerimislaua küljele kallutamisel kustuma. Tavalise lüliti puhul vilkuma vaid nuppu all hoides. Võib katsetada ka igale LED-ile oma digitaalsisendina defineeritud lüliti lisamist.

3.3 RGB ledi ühendamine ja RGB koodide katsetamine analogWrite abil

Käesolevas punktis näidatakse, kuidas ühendada peatükis 2.1.2 lähemalt tutvustatud värvimuutvat RGB valgusdiodi ning värvide muutmist, kasutades selleks ise defineeritud abifunktsiooni ja analogWrite käsku.

Projekti koostamiseks läheb vaja:

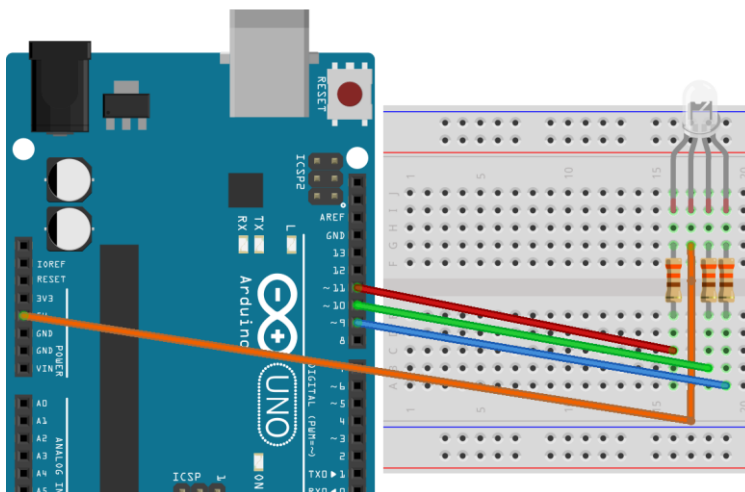
1. 1 x UNO mikrokontroller
2. 1 x USB kaabel
3. 1 x maketeerimislauad
4. 1 x RGB LED
5. 4 x tavalised üksikud isane-isane juhtmed
6. 3 x 330 Ω takisti

Enne komponentide ühendamist tuleb veenduda, et arendusplaat oleks arvuti küljest (ja muudest toidetest) lahti ühendatud.



Joonis 64. RGB LED-i jalgade seletus

Algus sarnaneb jälle peatükis 2.6.1. tehtud LED-ide vilgutamise projektile ainult seekord tuleb ühendada LED-i kolm lühemat jalga sarnaselt üksikutele valgusdiodidele, kuna need tähtistavadki ükskuid põhivärvidega LED-e (värviselgitused näidatud eelneval joonisel 64), mille üheaegsel erineva heledusega näitamisel saame luua erinevad värvikombinatsioone. Pesadena tuleb kasutada digitaalpesasid, kus juures on # või ~ märgistus, need pesad võimaldavad pinge muutmist analogWrite käsu abil, selles näites võtame kasutusele 9, 10 ja 11. Kuna tegemist on ühise anoodiga tüüpi RBG LED-iga siis tuleb anood (kõige pikem jalg) ühendada toitega (5 V). Ühendused peaksid sarnanema järgneval joonisel 65 kujutatule.



Joonis 65. RGB valgusdiodi ühendamine

Nüüd saab avada arenduskeskkonna ja hakata katsetama analogWrite käsu abil värvimängude loomist. Ideid värvide kombineerimiseks saab näiteks leheküljelt: http://www.rapidtables.com/web/color/RGB_Color.htm .

Järgnevas koodiplokis võib näha näidiskoodi, kus defineeritakse abifunktsioon, mis võtab argumentideks põhivärvide tugevused ning segab neid analogWrite käsu abil RGB valgusdiodil. Abifunktsiooni kutsutakse iga sekundi tagant välja erinevate argumentidega, mille tulemuseks on RGB LED-i värvi muutumine 1 sekundi tagant.

```
//defineerime pesad LED-i jalale vastava värvi järgi
int punanepesa = 11;
int rohelinepesa = 10;
int sininepesa = 9;
int sekund = 1000; //defineerime sekundi millisekundites

void setup()
{
  //seadistame väljundina
```

```

pinMode(punanepesa, OUTPUT);
pinMode(sininepesa, OUTPUT);
pinMode(rohelinepesa, OUTPUT);
}

void loop()
{
    //kutsume välja abifunktsiooni ning punase/sinise/roheline tugevust
    reguleerides katsetame eri värve
    segaV2rve(255, 0, 0); //punane
    delay(sekund); //ootame sekundi
    segaV2rve(0, 255, 0); //roheline
    delay(sekund);
    segaV2rve(0, 0, 255); //sinine
    delay(sekund);
    segaV2rve(255,255,0); //kollane
    delay(sekund);
    segaV2rve(0,255,255); //aqua
    delay(sekund);
    segaV2rve(255,0,255); //fuksiaroosa
    delay(sekund);
    segaV2rve(0,128,128); //sinakasroheline
    delay(sekund);
    segaV2rve(255,255,255); //valge
    delay(sekund);
}
//abifunktsioon, millele anname argumendiks punase, roheline ja sinise
tugevuse vastavalt RGB tabelile
void segaV2rve(int punane, int roheline, int sinine)
{
    //kuna tegu ühise anoodiga RGB LED-iga, siis tuleb RGB värvikoodide
    abil soovitud värvi saamiseks argumentide väärtused vastupidiseks
    muuta
    punane = 255 - punane;
    roheline = 255 - roheline;
    sinine = 255 - sinine;
    //erinevate värvide saamiseks määrame milline LED millise tugevusega
    põleb, kasutades analogWrite käsku
    analogWrite(punanepesa, punane);
    analogWrite(rohelinepesa, roheline);
    analogWrite(sininepesa, sinine);
}

```

Joonis 66. Kood eri värvide kuvamiseks RBG valgusdioodil

Parema ettekujutuse parajasti põlevast värvist saab, kui LED-ile ümber panna valgest õhukesest A4 paberist toru. Hetkel tulevad värvid alati kindlas järjekorras, projekti huvitavamaks muutmiseks võib järgmise värvi valida juhuslikult, seades värvikombinatsioonid vastavusse kindlate arvudega ning genereerides random() käsu abil juhuslikke muutujaid. Inglisekeelne juhend Arduinoga juhuslike muutujate kasutamiseks asub [siin](#).

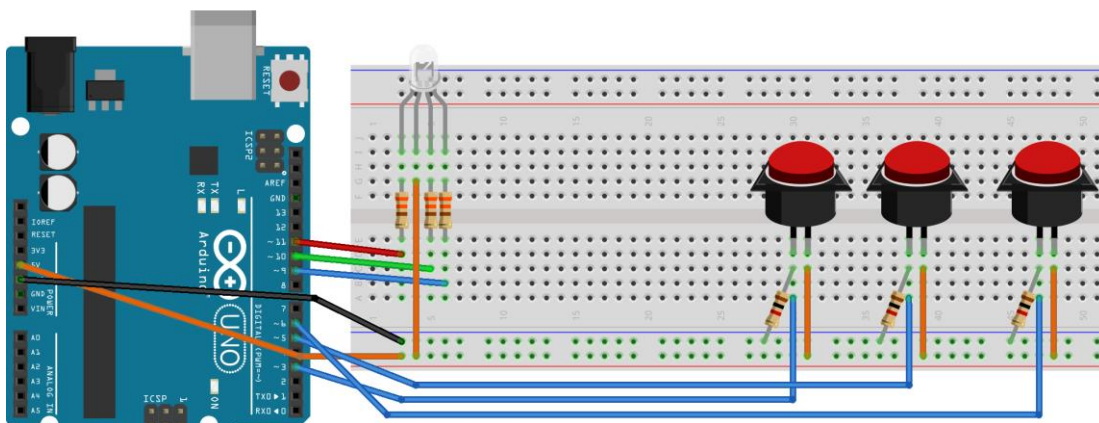
3.4 Kolme nupu abil RGB LEDi lülitamine

Eelmises alampeatükis tutvusime RGB LED-i ühendamise ja vilgutamisega, kuid meil polnud kontrolli parajasti põleva värvi üle. Selles alampeatükis selgitatakse, kuidas kontrollida RGB valgusdiodi kolme lüliti ja mitme IF-ELSE lause abil.

Projekti koostamiseks läheb vaja:

1. 1 x UNO mikrokontroller
2. 1 x USB kaabel
3. 1 x maketeerimislaud
4. 1 x RGB LED
5. 3 x 330 Ω takisti
6. 3 x 10 K Ω takisti
7. 3 x tavaline nupuga lüliti
8. 12 x tavalised üksikud isane-isane juhtmed

Ühendame RGB LED-i maketeerimislauale sarnaselt eelmisele alampeatükile. Kuna seekord on vaja paigutada ka kolm toidet vajavat lüliti, siis ühendame toitejuhtme maketeerimislauda äärmisele + märgistusega reale ning lisame LED-i toitmiseks ühe väikse juhtme juurde. Lülitite ühendumine sarnaneb alampeatüki 3.2 viimasele osale, kus tuli samuti ühendada lüliti maanduse, voolu ja digitaalpesaga, kasutades 10 kiloomist takistit. Ühendused kujutatud ka järgneval joonisel 67.



Joonis 67. RGB LED-i ja kolme nupu ühendumine

Nüüd tuleb nupud programmeerida nii, et iga nupp kontrolliks ühte värvi. Selleks kasutame eelmise alampeatüki LED-i värvide muutmise koodi kombineerituna alampeatüki 3.2 lõpus tutvustatud lüliti ja *digitalRead* käsu abil LED-i kontrollimise võttega. Näide tulemusest järgivas koodiplokis.

```
//defineerime pesad LED-i jalale vastava värvi järgi
int punanepesa = 11;
int rohelinepesa = 10;
int sininepesa = 9;
//defineerime lülitite pesad
int lyliti1 = 3;
int lyliti2 = 5;
int lyliti3 = 6;
//lülitamiste algväärtused
int on_lylitatud1 = 0;
int on_lylitatud2 = 0;
int on_lylitatud3 = 0;
//defineerime värvide algväärtused
int pu= 0;
int ro = 0;
int si = 0;

void setup()
{
  //seadistame väljundina
  pinMode(punanepesa, OUTPUT);
  pinMode(sininepesa, OUTPUT);
  pinMode(rohelinepesa, OUTPUT);
  //seadistame lülitid sisendina
  pinMode(lyliti1, INPUT);
  pinMode(lyliti2, INPUT);
  pinMode(lyliti3, INPUT);
}

void loop(){
  on_lylitatud1 = digitalRead(lyliti1);
  on_lylitatud2 = digitalRead(lyliti2);
  on_lylitatud3 = digitalRead(lyliti3);
  //kontrollime kas mõni nuppudest on vajutatud - kui on, siis lülitame
  nupule vastava värvi sisse
  if (on_lylitatud1 == HIGH){
    pu = 255;
  }
  else{
    pu = 0;
  }
  if (on_lylitatud2 == HIGH){
    ro = 255;
  }
  else{
    ro = 0;
  }
}
```

```

    }
    if (on_lylitatud3 == HIGH){
        si = 255;
    }
    else{
        si = 0;
    }
    //kutsume välja abifunktsiooni ning punase/sinise/roheline tugevust
    reguleerides katsetame eri värve
    segaV2rve(pu, ro, si);
}
//abifunktsioon, millele anname argumentiks punase, roheline ja sinise
tugevuse vastavalt RGB tabelile
void segaV2rve(int punane, int roheline, int sinine)
{
    //kuna tegu ühise anoodiga RGB LED-iga, siis tuleb RGB värvikoodide
    abil soovitud värvi saamiseks argumentide väärtused vastupidiseks
    muuta
    punane = 255 - punane;
    roheline = 255 - roheline;
    sinine = 255 - sinine;
    //erinevate värvide saamiseks määrame milline LED millise tugevusega
    põleb, kasutades analogWrite käsku
    analogWrite(punanepesa, punane);
    analogWrite(rohelinepesa, roheline);
    analogWrite(sininepesa, sinine);
}

```

Joonis 68. Kood RGB LED-il värvide muutmiseks kolme nupu abil

Nüüd peaks iga lüliti kontrollima erinevat RGB valgusdiodi värvi. Erinevaid nuppe üheaegselt all hoides saame värvikooslusi ise kontrollida. Hetkel põlevad LED-id ainult siis, kui nupud on all, edasiarendusena võib proovida nuppe programmeerida nii, et nupuvajutus jätab LED-i põlema nii kauaks kui seda värvi kontrollivat nuppu uuesti vajutatakse.

3.5 Ekraani ühendamine ja teksti kuvamine 1602 LCD ekraanil

Käesolevas punktis koostatakse juhend peatükis 2.1.7 tutvustatud 1602 LCD ekraani ühendamiseks, näidatakse ning kuidas sellel infot kuvada. Tausta ja teksti kontrastsuse reguleerimiseks kasutatakse potentsiomeetrit.

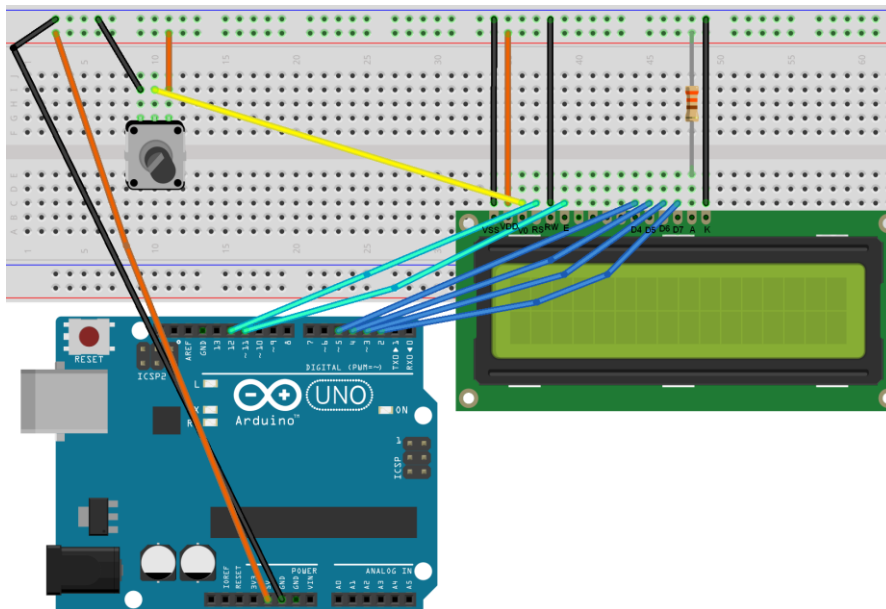
Projekti koostamiseks läheb vaja:

1. 1 x UNO mikrokontroller
2. 1 x USB kaabel

3. 1 x maketeerimislaud
4. 1 x 1602 LCD ekraan
5. 1 x potentsiomeeter
6. 15 x tavalised üksikud isane-isane juhtmed (6 lühemat ja 9 pikemat)
7. 1 x 330 Ω takisti

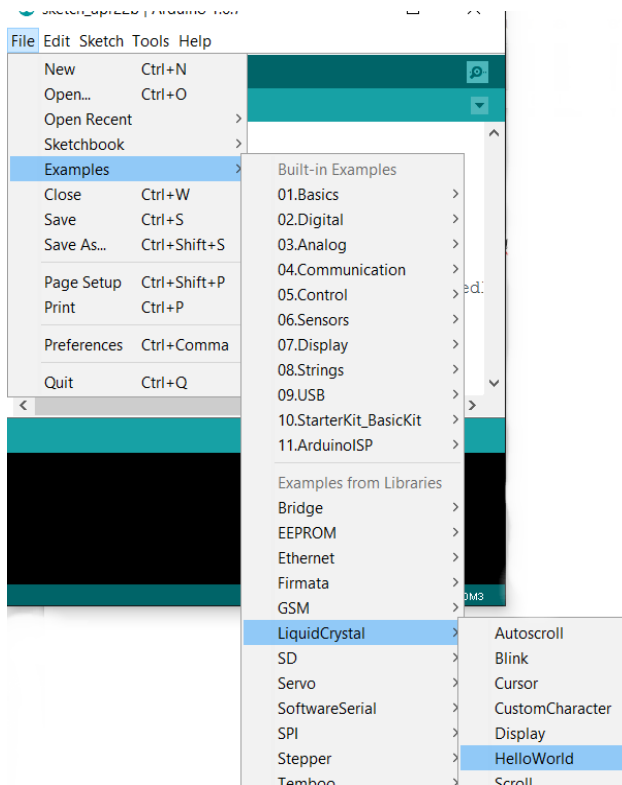
Enne komponentide ühendamist tuleb veenduda, et arendusplaat oleks arvuti küljest (ja muudest toidetest) lahti ühendatud.

Kõigepealt tuleb paigutada ekraan ja potentsiomeeter maketeerimislauale ja ühendada need, sarnaselt joonisel 69 kujutatule arendusplaadiga nii, et V0, mis määrab kontrasti oleks ühendatud potentsiomeeter, RS registrivalikuks digitaalpesaga 12, RW maandusega, kuna lugemist pole selles ülesandes vaja. Kirjutamise lubamiseks ühendada E digitaalpesaga 11. Ainult sümbolite ekraanile kuvamiseks 4-bitisest režiimist. Selleks ühendame pesad D4 -D7 digitaalpesadega 5-2. Mustad juhtmed tähistavad maandust, oranžid 5 V toidet. Taustavalgustuse anoodi (A) toidega ühendamisel kasutatakse takistit.



Joonis 69. LCD ekraani ja potentsiomeetri ühendamine.

Nüüd võib arendusplaadi USB juhtme abil arvutiga ühendada ning testida ekraani arenduskeskkonna näidisprogrammiga. Arenduskeskkonna LCD ekraani testimiseks mõeldud näited asuvad *File -> Examples -> LiquidCrystal* valime nendest näiteks “Hello World” (Joonis 70).



Joonis 70. LCD ekraani näitekoodi avamine

Seejärel leida koodi `setup ()` meetodist üles rida `lcd.print("hello, world!");` ning asendada sõna *world* enda nimega. Peale koodi plaadile laadimist peaks ekraani ülemisele reale ilmuma kiri “hello, *teie nimi*” ning alumisele sekundiloendur. Keerata potentsiomeetrit nii palju vasakule kui võimalik, et tagada maksimaalne kontrastsus. Teksti mitteilmumise korral peaks üle kontrollima potentsiomeetri ühendused, taustavalguse probleemide korral kahe viimane pesa (A ja K) ühendused.

Esimene katsetus tehtud, saab proovida ekraanile joonistada endale meelepärast liikuvat pilti. Selleks avada erisümbolite näide *File -> Examples -> LiquidCrystal -> CustomCharacter*, mille katsetamisel ilmub käsi üles-alla liigutav mehike. Ekraan ise koosneb 32 kastikesest (16 veergu 2 rida), millest igaüks on 5 x 8 pikslit. Kujutiste kuvamine toimub 8-baitiste järjendite abil, kus iga baiti 5 parempoolset bitti defineerivad ühe pikslirea [63]. Järgnevas koodiplokis on toodud CustomCharacker koodist inspireeritud liikuva pildi kood.

```
// lisame LiquidCrystal library
#include <LiquidCrystal.h>
// määrame, milliste digitaalpesadega ekraan ühendatud on
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
// loome erisümbolid
```

```

byte smiley[8] =
{0b00000,0b00000,0b01010,0b00000,0b00000,0b10001,0b01110,0b00000};
byte armsDown[8] =
{0b00100,0b01010,0b00100,0b00100,0b01110,0b10101,0b00100,0b01010};
byte armsUp[8] =
{0b00100,0b01010,0b00100,0b10101,0b01110,0b00100,0b00100,0b01010};
byte kass[8] =
{0b00000,0b00000,0b00000,0b10000,0b10011,0b11111,0b01110,0b01010};
byte tydruk[8] =
{0b00100,0b01010,0b00100,0b11111,0b00100,0b00100,0b01110,0b11111};
byte v2ikesyda[8] =
{0b00000,0b00000,0b00000,0b11011,0b11111,0b11111,0b01110,0b00100};

void setup() {
  // algväärtustatakse LDC ekraani read ja veerud
  lcd.begin(16, 2);
  // defineerime erisümbolid
  lcd.createChar(1, smiley);
  lcd.createChar(3, armsDown);
  lcd.createChar(4, armsUp);
  lcd.createChar(5, kass);
  lcd.createChar(6, tydruk);
  lcd.createChar(7, v2ikesyda);
}

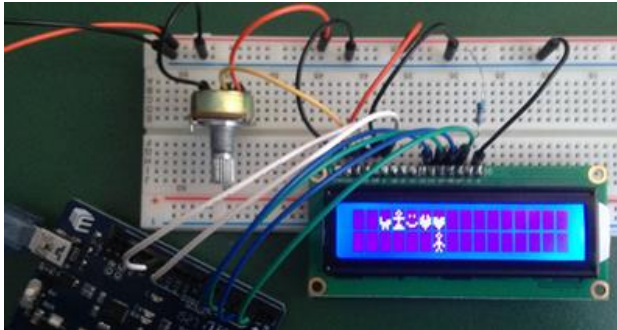
void loop() {
  // koht kuhu hakkame joonistama - 3. veerg, esimene rida
  lcd.setCursor(2, 0);
  // joonistame kassi
  lcd.write(5);
  // ja tüdruku
  lcd.write(6);
  //muudame kursori asukohta: 2. rida 7. veerg
  lcd.setCursor(6, 1);
  // joonistame poisi
  lcd.write(3);
  //ootame natuke
  delay(400);
  //muudame kursori asukohta: 1. rida 5. veerg
  lcd.setCursor(4, 0);
  //joonistame for tsükli abil väikese pausiga 3 smaili
  for (int koht = 0; koht < 3; koht++) {
    lcd.write(1);
    delay(400);
  }
  //muudame kursori asukohta: 2. rida 7. veerg
  lcd.setCursor(6, 1);
  lcd.write(4);
  //loobime südameid
  for (int koht = 6; koht >= 4; koht--) {
    lcd.setCursor(koht, 0);
    lcd.write(7);
    delay(400);
  }
}

```

```
}  
  delay(400);  
}
```

Joonis 71. Kood LCD ekaanile väikse liikuva pildi loomiseks

Tulemuseks on liikuv pilt (Joonis 72), kus poiss ja tüdruk üksteiseke südameid ja naerunägusid saadavad.



Joonis 72. Hetk liikuvast pildist LCD ekraanil.

Näidet saab muuta nii, et tekiks soovitud temaatikaga lugu. Piltide ekraanil kuvatavaks koodiks muutmisel aitab näiteks selline lehekülg: <https://omerk.github.io/lcdchangen/>. Hetkel liigub pilt automaatselt ühtlase kiirusega kuni arenduslaadi toitest eemaldamiseni, lugu saaks huvitavamaks programmeerida, lisades reaktsioonid näiteks nupuvajutustele või anduritega saadavatele mõõtmistulemustele.

3.6 Helianduri ühendamine, LEDide vilgutamine vastavalt müratugevusele

Käesolevas alampeatükis näidatakse, kuidas ühendada heliandurit ja programmeerida valgusdioode müratugevusele reageerima.

Projekti koostamiseks läheb vaja:

1. 1 x UNO mikrokontroller
2. 1 x USB kaabel
3. 1 x maketeerimislaud
4. 3 x vabalt valitud värvi LED-id
5. 3 x 330 Ω takistid
6. 1 x heliandur (*Sound sensor*)
7. 4 x tavalised üksikud isane-isane juhtmed

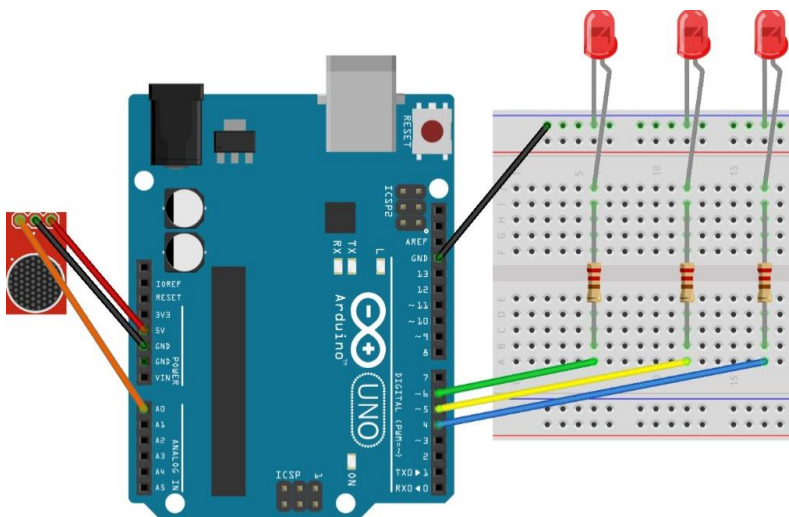
- 3 x isane -emane juhtmed või 1x 10 soonega isane - emane kaabel, millest kasutame kolme pesa.

Enne komponentide ühendamist tuleb veenduda, et arendusplaat oleks arvuti küljest (ja muudest toidetest) lahti ühendatud.

Algus sarnaneb peatükis 2.6.1. tehtud LED-ide vilgutamise projektile, tuleb ühendada kolm LED-i rööpühenduses maketeerimislauale nii, et katood (LED-i lühem jalg) oleks ühendatud maketeerimislaua miinusreale ning anood (LED-i pikem jalg) ühendatud maketeerimilaua keskel takistiga ning seejärel digitaalsignaali, antud näites on kasutusel digitaalpesad numbritega 4, 5 ja 6. Miinusreale tuleb lisada ka maandusjuhe (ühendatud digitaalpesade kohal olevase GND pesasse).

Nüüd kui LED-id ühendatud, tuleb paika panna peatükis 2.1 tutvustatud heliandur. Selleks tuleb vaadata helianduril olevaid märgistusi ja ühendada vastavalt sellele isane - emane juhtmed helianduri ja arendusplaadi vahel järgnevalt: 5 V toitega, GND maandusega, A alanoogsisendiga (antud näites kasutusel analoogpesa A0) ning D võib hetkel ühendamata jätta, kuna selles projektis me heliandurile digitaalühendust ei vaja.

Kui kõik ühendused tehtud, siis võib need üle kontrollida jooniselt 73. Joonisel kujutatud helianduril on ainult kolm jalga, aga see ei muuda meie skeemi, kuna jooniselt puuduv digitaalväljund pole kasutusel.



Joonis 73. Helianduri ja kolme LED-i ühendamine

Nüüd on aeg ühendada arendusplaat USB juhtme abil arvutiga ning avada arenduskeskkond, teha uus koodiaken ning kirjutada sinna järgnev kood (kommentaariid võib ära jätta).

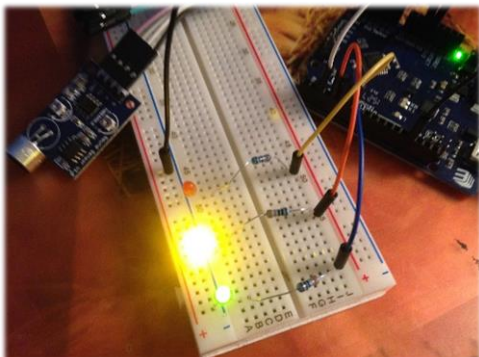
```
// määrame algväärtused
int heliandur = A0;
int kynnis = 532;
int sensori_v22rtus = 0;

void setup() {
    //defineetime digitaalpesad, kuhu LED-id on ühendatud
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
}

void loop() {
    //loeme müra väärtust
    sensori_v22rtus = analogRead(heliandur);
    //kontrollime kas müra väärtus ületab konstandina antud künnise
    if (sensori_v22rtus >= kynnis) {
        //kui jah, siis süütame ühe LED-i ja jätame korraks põlema
        digitalWrite(4, HIGH);
        delay(200);
    }
    //kontrollime uusti kas müra väärtus ületab 1.1 kordse konstandina
    antud künnise
    if (sensori_v22rtus >= (1.1*kynnis)) {
        //kui jah, siis süütame järgmise LED-i ja jätame korraks põlema
        digitalWrite(5, HIGH);
        delay(400);
    }
    //kontrollime uusti kas müra väärtus ületab 1.2 kordse konstandina
    antud künnise
    if (sensori_v22rtus >= (1.2*kynnis)) {
        // kui jah, siis süütame järgmise LED-i ja jätame korraks põlema
        digitalWrite(6, HIGH);
        delay(600);
    }
    //vaastasel juhul kustutame LED-id
    else {
        digitalWrite(4, LOW);
        digitalWrite(5, LOW);
        digitalWrite(6, LOW);
    }
}
```

Joonis 74. Helile reageerivate valgusdioodide kood

Seejärel laadida kood plaadile ning testida projekti näiteks heliandurisse lauldes, muusikat lastes, puhudes või selle läheduses koputades (Joonis 75). Väiksema müra peale peaks süttima neljandasse pesase ühendatud LED, natuke suurema müra peale 5. ja veel suurema müra peale ka viimane.



Joonis 75. Helile reageerivate valgusdiodide testimine

Vaikuse korral peaksid kõik tuled korraga kustuma. Edasiarendusena võib proovida sama projekti suurema arvu valgusdiodidega.

3.7 Fototakisti abil valguse mõõtmine, Serial monitoriga tutvumine, valgustundlik LED

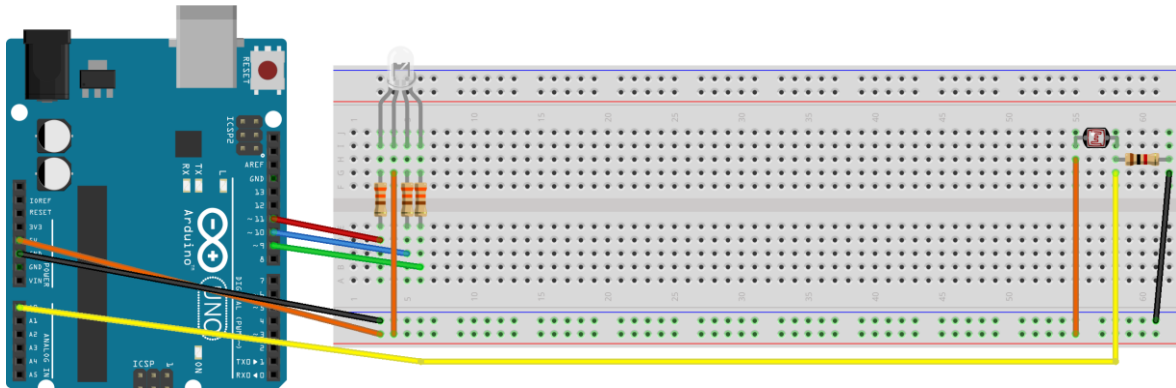
Käesolevas punktis koostatakse juhend peatükis 2.1.8 tutvustatud fototakisti ühendamiseks. Näidatakse, kuidas mõõdetud analoogsignaali digitaalseks muuta ning Serial Monitori kaudu kuvada. Saadud tulemusi kasutades proovime seada RGB LED-i valgusele võimalikult silmasõbralikult reageerima.

Projekti koostamiseks läheb vaja:

1. 1 x UNO mikrokontroller
2. 1 x USB kaabel
3. 1 x maketeerimislaud
4. 1 x fototakisti
5. 1 x 10 K Ω takisti
6. 1 x RGB LED
7. 3 x 330 Ω takisti

8. 8 x tavalised üksikud isane-isane juhtmed

Ühendada kõigepealt maketeerimislaua ühte äärde sarnaselt järgneva joonisega fototakisti nii et üks jalg oleks ühendatud toite ning teine läbi maandusesse mineva takisti analoogsignaali. Nüüd lisada maketeerimislaua teise otsa, et LED-i valgus ei mõjutaks andurit, alampeatükkidest 3.2 ja 3.3 tuttav RGB LED.



Joonis 76. RGB valgusdiodi ja fototakisti ühendamine

Kuna me ei tea milliste tulemustega fototakisti valgusele reageerib, siis testime esialgu ainult fototakistit, kasutades selleks järgnevas koodiplokis kujutatud AnalogInOut (File- > Examples -> Analog -> AnalogInOut) näitel põhinevat koodijuppi, kus kirjutame väärtused iga sekundi tagant Serial Monitori.

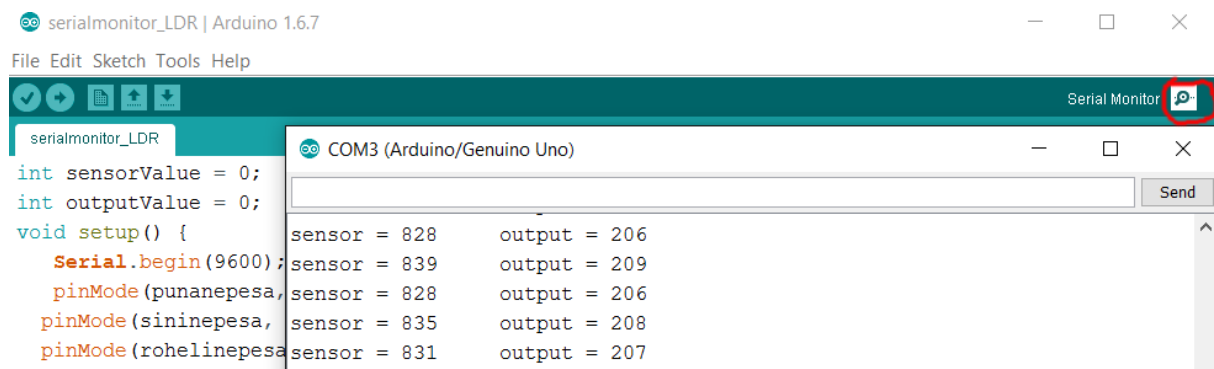
```
int LDR = 0; //defineerime analoogseda, kuhu fototakisti on ühendatud
//algväärtused
int sensorValue = 0;
int outputValue = 0;
void setup() {
  Serial.begin(9600); //Serial monitori seadistamine
}

void loop() {
  sensorValue = analogRead(LDR); //loeme mõõdetud analoogväärtuse
  outputValue = map(sensorValue, 0, 1023, 0, 255); //kaardistame
  selle analoogväljundi vahemikku

  // prindime tulemused Serial Monitori
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);
  delay(1000); // ootame sekundi
}
```

Joonis 77. Kood fototakisti väärtuste Serial Monitori kirjutamiseks

Monitori jälgimiseks avada see arenduskeskkonna ülemise rohelise riba paremas ääres asuvast luubi ikoonist (Joonis 78).



Joonis 78. Serial Monitori avamine

Alampeatükkides 3.2 ja 3.3 selgitatud värvide segamist ja fototakisti abil saadud andmete analüüsi kombineerides saame koodi, mis reguleerib RGB LED-i värvust valguse vähenedes eredalt sinakasrohekalt soojemaks punakamaks ning prindib info valguse tugevuse kohta Serial Monitori.

```
const int LDR = 0; //defineerime analoogpesa, kuhu fototakisti on
ühendatud
int punanepesa = 11;
int rohelinepesa = 10;
int sininepesa = 9;

//algväärtused
int sensorValue = 0;
int outputValue = 0;
void setup() {
  Serial.begin(9600); //Serial monitori seadistamine
  pinMode(punanepesa, OUTPUT);
  pinMode(sininepesa, OUTPUT);
  pinMode(rohelinepesa, OUTPUT);
}
void loop() {
  sensorValue = analogRead(LDR); //loeme mõõdetud analoogväärtuse
  outputValue = map(sensorValue, 0, 1023, 0, 255); //kaardistame
selle analoogväljundi vahemikku
  // prindime tulemused Serial Monitori
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);
  segaV2rve((255-(outputValue/2)),outputValue,outputValue);
  //reguleerib värvide kontrasti vastavalt signaalile
  delay(1000); // ootame sekundi
}
void segaV2rve(int punane, int roheline, int sinine)
```



```
{
  punane = 255 - punane;
  roheline = 255 - roheline;
  sinine = 255 - sinine;
  analogWrite(punanepesa, punane);
  analogWrite(rohelinepesa, roheline);
  analogWrite(sininepesa, sinine);
}
```

Joonis 79. Kood RBG LED-i värvi muutmiseks vastavalt valgustugevusele

Testimiseks on hea hoida andurit korraks pimedas ning seejärel proovida otsese ereda valguse peale laskmist näiteks taskulambi abil. Edasiarendusena võib proovida mõõtmistulemuste kuvamist LDC ekraanil nii et ekraani taustavalgus reageeriks nähtava valguse eredusele.

3.8 Juhtkangi ühendamise ja mootorite juhtimine

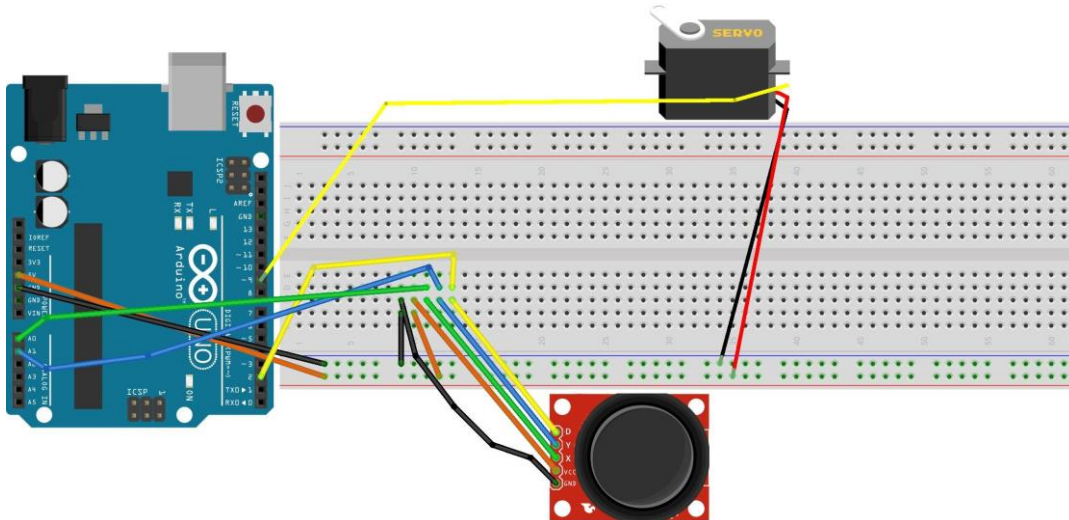
Käesolevas alampeatükis ühendatakse peatükis 2.1.6 tutvustatud mootoreid ning näidatakse, kuidas neid juhtkangi abil juhtida. Kasutusel on 180 kraadi pöörav servo mootor, mida saab juhtida näiteks lülitile vajutama ning samm-mootor, mille liikumiskiiruse üle omame kontrolli.

Projekti koostamiseks läheb vaja:

1. 1 x UNO mikrokontroller
2. 1 x USB kaabel
3. 1 x maketeerimislaud
4. 1 x juhtkang
5. 1 x 180 kraadine servo mootor
6. 1 x samm-mootor
7. 1 x samm-mootori draiver
8. 15 x tavalised üksikud isane-isane juhtmed

Kõigepealt ühendame mootori ja juhtkangi sarnaselt joonisel 80 kujutatule makereerimislauda vahendusel arendusplaadiga. Kuna meie juhtkang on jalgedega, siis võib selle otse maketeerimislauale asetada, joonisel on jalgade asemel juhtmed, mis ei muuda ühenduste loogikat. Juhtkangi ühendamisel tuleb jälgida pesade märgistusi, kuna pesade järjekord võib

erinevate tootjate toodetel olla erinev. Käesoleva komplekti tootel on maandus alumine (GND), järgmine 5 V toide (VCC), x- koordinaati tähtistav X ja y- koordinaat Y ühendatakse analoogpesadega ning digitaalsignaali (D) tuleb ühendada digitaalpesaga. Servomootoril kirjalikud märgistused puuduvad, kuid sellest tulevad juhtmed on otstarbele kohaste värvidega (must maandus, punane toide ning kollakasoranžikas digitaalsignaali).



Joonis 80. Servo mootori ja juhkangi ühendamine

Katsetame areduskeskkonnaga kaasas oleva näitekoordi abil ainult mootori liikumist, selleks avame ja laadime plaadile näidiskoodi Sweep (File ->Examples ->Servo ->Sweep), mis liigutab mootorit ühele ja teisele poole.

Nüüd, kui nähtud kuidas mootor töötab, saab juurde kirjutada kangist juhtimise osa. Järgnevas koodiplokis kasutame 180 kraadise servo liigutamiseks ainult juhtkangi x-telge, alguses on meil defineeritud ka y pesa, see leiab kasutust järgmises ülesandes. Telgede väärtuste vaatamiseks kasutame alampeatükis 3.7 lähemalt tutvustatud Serial Monitori.

```
// By Explore Labs

#include <Servo.h> //servo lisad, mis lihtsustavad juhtimist
Servo servo; // servoobjekt servomootori kontrollimiseks

int joyX = A0; // juhtkangi X-teljega ühendatud analoogpesa
int joyY = A1; // juhtkangi Y-teljega ühendatud analoogpesa
int x, y; // muutujad analoogpesadest vastuste lugemiseks

void setup()
{
  servo.attach(9); // defineerime servo pini
  Serial.begin(9600); //defineerime Serial Monitori, et näha infot
  liikumise kohta
}
```

```

}

void loop()
{
  x = map(analogRead(joyX), 0, 1023, 900, 2100); // loeme analoogpesast
  x -telje väärtuse ja paneme selle servo jaoks sobivale skaalale
  //prindime vastuse kontrollimise eesmärgil serial monitorile
  Serial.print("x-telg: ");
  Serial.println(x);

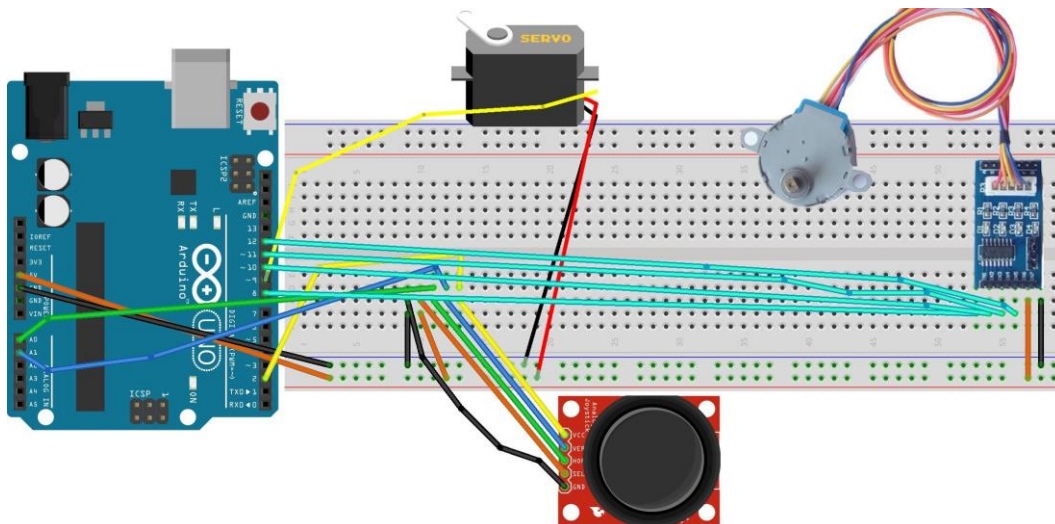
  servo.write(x); // anname servole x telje väärtuse
  delay(15); // ootame servo liikumist
}

```

Joonis 81. Kood servo mootori juhtimiseks juhtkangiga [64]

Nüüd peaks võimalik olema juhtkangist servo mootorit edasi-tagasi liigutada, selline funktsionaalsus on kasulik näiteks eemalt lüliti või mõne nupu klõpsamiseks. Hetkel oli y-telg defineeritud, aga väärtusi sealt ei loetud, kuna ühe mootori juhtimiseks piisab ühest teljest.

Kasutamaks suuremat osa juhtkangi võimalustest, ühendame maketeerimislauale lisaks ühe samm-mootori. Ühendamine on mingil määral sarnane servo mootori ühendamisele, kuid erinevalt servost tuleb vahele lisada ka draiver nii, et mootorist tulevad juhtmed ühendatakse sarnaselt joonisel 82 kujutatule draiveri külge ning draiver omakorda maketeerimislauale. Juhtmed tuleb ühendada vastavalt draiveril olevatele märgistustele: GND maandusega, 5V toitega ning ülejäänud digitaalpesadega.



Joonis 82. Juhtkangi, servo ja samm-mootori korraga ühendamine

Täiustame eelmist koodi nii, et samm-mootori kiirus sõltuks juhtkangi y-teljest.

```

/*
  Stepper Motor Control - speed control
  Created 30 Nov. 2009

```

```

Modified 28 Oct 2010
by Tom Igoe
Servo osa- By Explore Labs

*/
//servo ja stepperi lisad, mis lihtsustavad nende mootorite juhtimist
#include <Servo.h>
#include <Stepper.h>

const int stepsPerRevolution = 200; //sõltub mootori tüübist, antud
mootor töötab 200 spr
Stepper myStepper(stepsPerRevolution, 8, 10, 11,12); //defineerime
sammootori pinid, pinide valik selline, kuna 9. oli juba servo jaoks
kasutusel
Servo servo; // servoobjekt servomootori kontrollimiseks

int joyX = A0; // juhtkangi X-teljega ühendatud analoogpesa
int joyY = A1; // juhtkangi Y-teljega ühendatud analoogpesa
int x, y; // muutujad analoogpesadest vastuste lugemiseks

void setup()
{
  servo.attach(9); // defineerime servo pini
  Serial.begin(9600); //defineerime Serial Monitori, et näha infot
liikumise kohta
}

void loop()
{
  x = map(analogRead(joyX), 0, 1023, 900, 2100); // loeme analoogpesast
x -telje väärtuse ja paneme selle servo jaoks sobivale skaalale
  y = map(analogRead(joyY), 0, 1023, 0, 100); // loeme analoogpesast y -
telje väärtuse ja paneme selle sammootori jaoks sobivale skaalale
  //prindime vastused kontrollimise eesmärgil serial monitorile
  Serial.print("x-telg: ");
  Serial.print(x);
  Serial.print("y-telg: ");
  Serial.println(y);

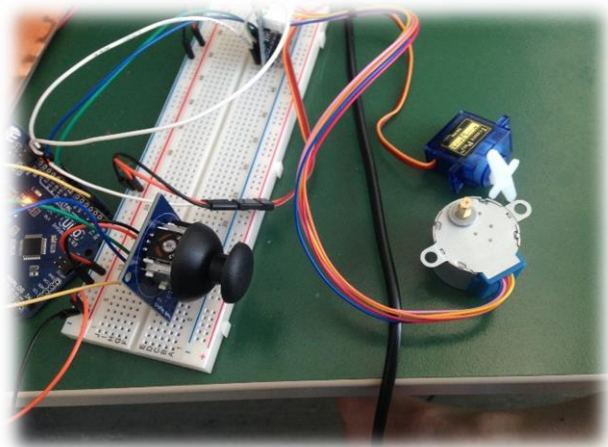
  if (y > 0) {
    myStepper.setSpeed(y/2); //muudame sammootori kiirust vastavalt
juhtkangi y telje väärtusele
    myStepper.step(stepsPerRevolution / 100); // liikutame sammootorit
  }

  servo.write(x); // anname servole x telje väärtuse
  delay(15); // ootame servo liikumist
}

```

Joonis 83. Servo ja samm-mootori korraga juhtimine [64, 65]

Nüüd saab juhtkangi abil korraga juhtida mõlemat mootorit (Joonis 84).



Joonis 84. Juhtkang ja mootorid

Parema ettekujutuse sammootori liikumiskiirusest saab, kui selle külge kinnitada sobiv ratas, ratta puudumisel võib liikuva osa külge paberist väikese lipukese meisterdada.

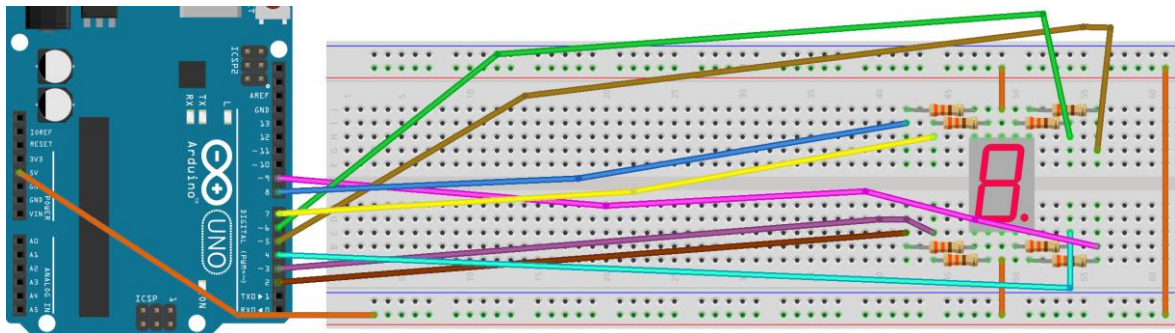
3.9 Infrapuna vastuvõtja, pult ja 7-segmeniline ühe numbriga LED

Käesolevas alampeatükis näidatakse, kuidas ühendada peatüki 2.1 alampunktides tutvustatud infrapuna vastuvõtjat ja 7-segmenilist 1-numbrilist valgusdiodi. Tänu infrapuna vastuvõtjale saame LED-i eemalt puldiga kontrollida, kas käivitada LED-il jooksvat numbrite tsüklit või kuvada valgusdiodil hoopis puldil vajutatud numbrit. Puldilt tulevate signaalide vahetamiseks sobib hästi programmeerimises tuntud SWITCH - lause kasutamine.

Projekti koostamiseks läheb vaja:

1. 1 x UNO mikrokontroller
2. 1 x USB kaabel
3. 1 x maketeerimislaud
4. 1 x 7-segmeniline ühise anoodiga numberLED
5. 1 x pult
6. 1 x infrapuna vastuvõtja
7. 8 x 330 oom takisti
8. 16 x tavalised üksikud isane-isane juhtmed

Ühendame kõigepealt 7-segmenndilise LED-i sarnaselt joonisel 85 kujutatule maketeerimislauale nii, et iga digitaalpesaga ühenduse vahel oleks takisti. Kuna tegu on ühise anoodiga LED-iga, siis tuleb pesad 3 ja 8 ühendada toitega.



Joonis 85. 7-segmenndilise valusdiodi ühendamine

Proovime sellel järgneva koodi abil numbreid kujutada ja eelmistest alampeatükkidest tuttava FOR tsükli abil neid üheksast nullini lugema panna.

```
// Arduino 7 segment display example software
// http://www.hacktronics.com/Tutorials/arduino-and-7-segment-led.html
// License: http://www.opensource.org/licenses/mit-license.php
// defineeritakse numbrite koostamiseks vajalikud bitikombikatsioonid
//kuna tegu ühise anoodiga, siis 1 tähendab välja ja 0 sisse lülitatud
byte seitsme_seg_numbrid[10][7] = { { 0,0,0,0,0,0,1 }, // = 0
                                     { 1,1,0,0,1,1,1 }, // = 1
                                     { 0,0,1,0,0,1,0 }, // = 2
                                     { 1,0,0,0,0,1,0 }, // = 3
                                     { 1,1,0,0,1,0,0 }, // = 4
                                     { 1,0,0,1,0,0,0 }, // = 5
                                     { 0,0,0,1,0,0,0 }, // = 6
                                     { 1,1,0,0,0,1,1 }, // = 7
                                     { 0,0,0,0,0,0,0 }, // = 8
                                     { 1,0,0,0,0,0,0 } // = 9
                                   };

void setup() { //seadistame inid 2-8 väljunditena
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  lisaPunkt(0); //lülitame punkti ka sisse
}
//funktsioon punkti sisse lülitamiseks
void lisaPunkt(byte punkt) {
  digitalWrite(9, punkt); //punkt on ühendatud digitaalpesasse 9
}
```

```

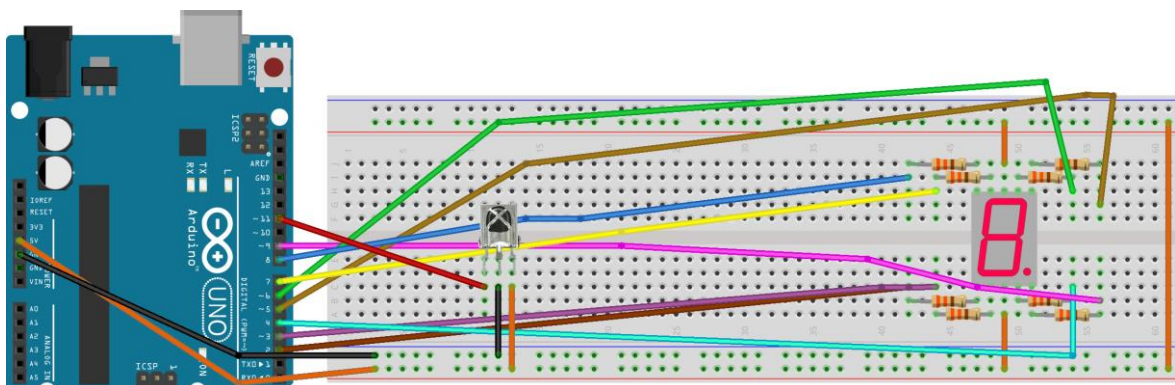
//numbrite kirjutamise funktsioon, argumendiks võetakse kirjutatav
number
void kirjutaNumbrid(byte arv) {
  byte pesa = 2; //esimene pesa, mis digitaalpesadest LED-il kasutusel
  on
  //tsükkel numbrite kirjutamiseks eelnevalt defineeritud muutuja abil
  for (byte segLoendur = 0; segLoendur < 7; ++segLoendur) {
    digitalWrite(pesa, seitsme_seg_numbrid[arv][segLoendur]);
    ++pesa;
  }
}

void loop() {
  //tsükkel numbrite 9 kuni 0 lugemiseks 1 sekundise vahega
  for (byte loendur = 10; loendur > 0; --loendur) {
    delay(1000);
    kirjutaNumbrid(loendur - 1);
  }
  delay(4000); //tsükki lõppedes tehakse 4 sekundine paus
}

```

Joonis 86. Kood numbrite kuvamiseks 7-segmendilisel valgusdioodil [28]

Hetkel töötab programm kohe arendusplaadi toitega ühendamise hetkest ning tsükliina pidevalt uuesti kuni toite eemaldamiseni. Loenduri üle suurema kontrolli saavutamiseks ühendame lisaks ka infrapuna vastuvõtja, mis annab meile võimaluse koodi puldi abil käivitada või lihtsalt enda soovitud numbreid kuvada. Vastuvõtja tuleb ühendada, sarnaselt järgneval jonisel kujutatule, maanduse toite ja ühe digitaalpesaga, selles näites kasutatud pesa number 11.



Joonis 87. Infrapuna vastuvõtja ühendamine koos 7-segmendilise LED-iga

Lisame puldiga juhitavuse osa eelnevale koodile juurde. Kõigepealt peame ära asendama ka ühe teegi, kuna Arduino-ga kaasas olev RobotIRremote library tekitab konflikti. Selleks tuleb tõmmata *Infrared remote library* aadressilt: <https://github.com/z3t0/Arduino-IRremote> (leheltpärit Arduino-IRremote-master.zip fail lisatud ka käesoleva töö lisadesse) ning asendada sellega *Arduino libraries* kaustas (tavaliselt *C:\Program Files (x86)\Arduino\libraries*) asuv RobotIRremote kaust [66].

Nüüd peaks IRremote teek ilusti töötama ja saame proovida 7-segmendilist LED-i puldist lülitatavaks muuta. Selleks lisame eelmisele koodile IR vastuvõtja seadistamiseks vajalikud muutujad ja puldist tulevatele signaalidele reageerimise osa. Selleks kasutame *SWITCH*-lauset, kus ühele sündmusele vastava iga erineva juhu (*case*) korral tehakse mingi tegevus, kui ükski defineeritud juht ei esine, siis saadetakse korda *default* plokis olev tegevus.

```
// LED-i osa: Copyright Arduino 7 segment display example software
// http://www.hacktronics.com/Tutorials/arduino-and-7-segment-led.html
// License: http://www.opensource.org/licenses/mit-license.php
// IR vastuvõtja osa: Copyright https://brainy-bits.com/tutorials/ir-remote-arduino/
#include "IRremote.h" //teek infrapunasiisiga toimetamiseks

int RECV_PIN = 11; //defineerime digitaapesa, millega vastuvõtja on
ühendatud

IRrecv irrecv(RECV_PIN); // loome vastuvõtja muutja
decode_results results; // vastuse muutuja

//                                pesad: 2,3,4,5,6,7,8
byte seitsme_seg_numbrid[10][7] = { { 0,0,0,0,0,0,1 }, // = 0
                                     { 1,1,0,0,1,1,1 }, // = 1
                                     { 0,0,1,0,0,1,0 }, // = 2
                                     { 1,0,0,0,0,1,0 }, // = 3
                                     { 1,1,0,0,1,0,0 }, // = 4
                                     { 1,0,0,1,0,0,0 }, // = 5
                                     { 0,0,0,1,0,0,0 }, // = 6
                                     { 1,1,0,0,0,1,1 }, // = 7
                                     { 0,0,0,0,0,0,0 }, // = 8
                                     { 1,0,0,0,0,0,0 } // = 9
                                   };

void setup() {
  Serial.begin(9600); //käivitame serial monitori
  irrecv.enableIRIn(); // käivitame vastuvõtja
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  writeDot(0);
}
void writeDot(byte dot) {
  digitalWrite(9, dot);
}

void kirjutaNumbrid(byte arv) {
```



```

    byte pesa = 2; //esimene pesa, mis digitaalpesadest LED-il kasutusel
on
//tsükkel numbrite kirjutamiseks eelnevalt defineeritud muutuja abil
    for (byte segLoendur = 0; segLoendur < 7; ++segLoendur) {
        digitalWrite(pesa, seitsme_seg_numbrid[arv][segLoendur]);
        ++pesa;
    }
}

void loop() {
    if (irrecv.decode(&results)) // kui saame infrapuna signaali
    {
        translateIR(); //kutsume välja funktsiooni, kus tuvastame signaali
        Serial.println(results.value, HEX); //prindime saadud signaali koodi
serial monitori
        irrecv.resume(); // ootame uut signaali
    }
}

//numbrite 9-0 lugemise funktsioon
void loe(){
    for (byte loendur = 10; loendur > 0; --loendur) {
        delay(1000);
        kirjutaNumbrid(loendur - 1);
    }
}

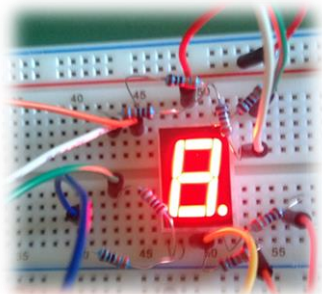
// IR käskude tõlkimise funktsioon
void translateIR() {
    //kasutame switch käsku, et vahetada tegevusi vastavalt puldi
signaalile, vastuse kirjutame ka serial monitori
    switch(results.value)
    {
        case 0xFF629D: Serial.println(" FORWARD"); break;
        case 0xFF22DD: Serial.println(" LEFT"); break;
        case 0xFF02FD: Serial.println(" -OK-"); loe(); break; //käivitame
numbrite lugemise 9-0
        case 0xFFC23D: Serial.println(" RIGHT"); break;
        case 0xFFA857: Serial.println(" REVERSE"); break;
        //numbriklahvide vajutamisel kuvame soovitud numbrid
        case 0xFF6897: Serial.println(" 1"); kirjutaNumbrid(1); break;
        case 0xFF9867: Serial.println(" 2"); kirjutaNumbrid(2); break;
        case 0xFFB04F: Serial.println(" 3"); kirjutaNumbrid(3); break;
        case 0xFF30CF: Serial.println(" 4"); kirjutaNumbrid(4); break;
        case 0xFF18E7: Serial.println(" 5"); kirjutaNumbrid(5); break;
        case 0xFF7A85: Serial.println(" 6"); kirjutaNumbrid(6); break;
        case 0xFF10EF: Serial.println(" 7"); kirjutaNumbrid(7); break;
        case 0xFF38C7: Serial.println(" 8"); kirjutaNumbrid(8); break;
        case 0xFF5AA5: Serial.println(" 9"); kirjutaNumbrid(9); break;
        case 0xFF42BD: Serial.println(" *"); break;
        case 0xFF4AB5: Serial.println(" 0"); kirjutaNumbrid(0); break;
        case 0xFF52AD: Serial.println(" #"); break;
        case 0xFFFFFFFF: Serial.println(" REPEAT");break;
        default:
        Serial.println(" muu nupp "); //tundmatu nupu korral
    }
}

```

```
}  
  delay(500);  
}
```

Joonis 88. 7-segmendilise LED-i juhtimine puldi abil [28-29]

Nüüd peaks LED puldil iga erineva numbriga peale kuvama valitud numbriga ning "OK" klõpsamise peale alustama numbriga üheksast alla lugemise tsükli.



Joonis 89. Number 8 kuvamine 7-segmendilisel näidikul

Erinevatel pultidel on signaali vastused erinevad, puldilt saadud signaal kirjutatakse ka alampeatükis 3.7 lähemalt tutvustatud Serial Monitori ning soovi korral on võimalik *case-*tingimusi muutes antud kood mõne teise puldi jaoks ümber programmeerida.

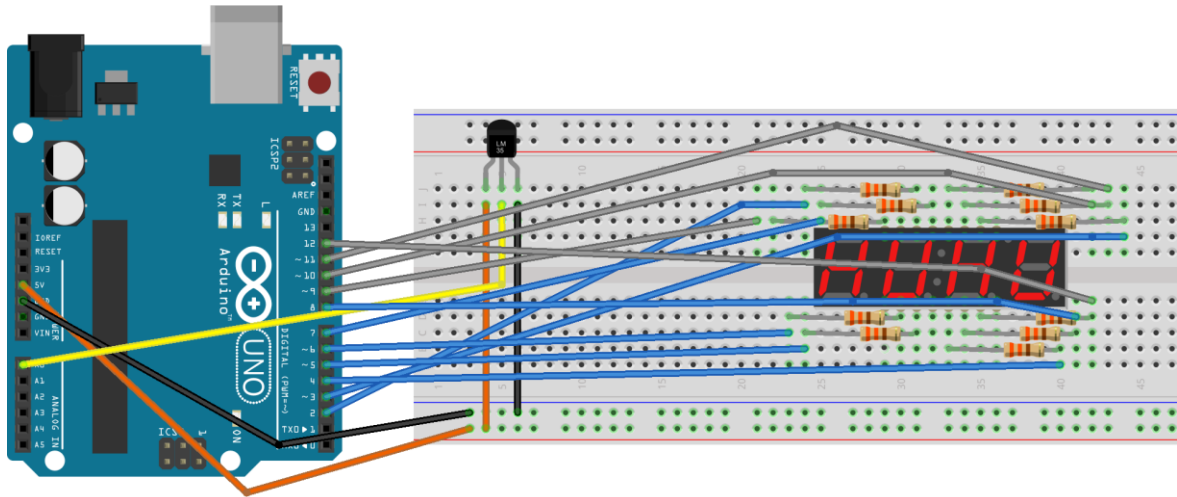
3.10 Temperatuuri kuvamine 4-kohalisel 7-segmendilisel näidikul

Käesolevas punktis näidatakse, kuidas ühendada peatüki 2.1 alampeatükkides tutvustatud temperatuuriandurit ja 7-segmendilist 4-numbrilist valgusdiodi. Koostatakse juhend temperatuuri mõõtmiseks LM35 temperatuurianduri abil ning kuvatakse saadud tulemused näidikule. Näidiku aja uuendamiseks soovitud ajaperioodi tagant kasutatakse *while*-tsükli.

Projekti koostamiseks läheb vaja:

1. 1x UNO mikrokontroller
2. 1 x USB kaabel
3. 1 x maketeerimislaud
4. 1 x 7-segmendiline 4-kohaline valgusdiod
5. 1 x temperatuuriandur
6. 16 x tavalised üksikud isane-isane juhtmed
7. 7 x 330 oom takisti

Ühendame kõigepealt 7-segmendilise valgusdiodi sarnaselt joonisel 90 kujutatule maketeerimislauale nii, et iga digitaalpesaga ühenduse vahel oleks takisti ning pesadevahelised ühendused arendusplaadiga vastaksid tabelile 5. Temperatuuriandur ühendatakse sarnaselt joonisele 90 nii, et lame külg jääb ette poole.



Joonis 90. 4-numbrilise 7-segmendilise LED-i ja temperatuurianduri ühendamine

7-segmendilise valgusdiodi jalgu tähistatakse numbritega 1-12, lugemist alustatakse alt vasakult (alumine vasakpoolne pesa on 1). Alumist äärt märgib sinna kirjutatud numbrikombinatsioon. Näidiku pesad, mis on ühendatud arendusplaadi pesadesse 2-8 määravad ära valgusdiodi piirkonnad, ülejäänud näitavad, millise näidiku LED-iga tegu.

Tabel 5. 4-numbrilise 7-segmendilise valgusdiodi ühendamine [67]

Pesa arendusplaadil	Pesa 7-segmendilisel näidikul	Täht LED-i skeemil, mis tähistab ühendatud kohta
2	11	A
3	7	B
4	4	C
5	2	D
6	1	E
7	10	F
8	5	G
9	12	Esimene LED (D1)
10	9	Teine LED (D2)
11	8	Kolmas LED (D3)
12	6	Neljas LED (D3)

Temperatuuriandur töötab sarnaselt varem kasutatud analooganduritega, kuid arusaadavate tulemuste saamiseks tuleb need teisendada kraadideks Celsiuse järgi [68]. Tulemuste vaatamiseks kirjutame need Serial Monitori.

7-segmendilisel näidikul põleb tegelikult korraga ainult üks sümbol, aga kuna kirjutatakse järjest 1 ms vahega, siis inimsilmale jääb mulje, et kõik numbrid põlevad korraga. Probleem tekib sellest, et näidikut uuendatakse liiga tihti, mille tõttu muutub väikese pideva temperatuurikõikumise tõttu viimane number loetamatuks. Selle vältimiseks tuleb lisada tulemuste näidikul kuvamise osale ümber while-tsüklitel, mille abil saab kirjutada näidikule soovitud aja jooksul viimati mõõdetud näitu. Nii suudame jätta mulje nagu uuendatakse näidikut soovitud ajaperioodi tagant. Temperatuuri mõõtmise ja tulemuste töötlemise osa peab jääma tsüklist välja.

```
/*
    A
   ---
  F |   | B
    | G |
    ---
  E |   | C
    |   |
    ---
    D
    This example code is in the public domain.
    Koodi allikad:
    http://duinos.net/files/2015/common anode 4digit 7 segment led displ
ay.ino
    https://www.youtube.com/watch?v=iZI1GjCvIiw
*/

//algväärtustame ajamuutujad while tsükli jaoks
int algus =0;
int lopp =0;
//digitaalpesad 2-12 on ühendatud 7-segmendilise valgusdiodiga
int pinA = 2;
int pinB = 3;
int pinC = 4;
int pinD = 5;
int pinE = 6;
int pinF = 7;
int pinG = 8;
int D1 = 9;
int D2 = 10;
int D3 = 11;
int D4 = 12;

void setup() {
  Serial.begin(9600); //Serial Monitori seadistamine
```

```

pinMode(0, INPUT); //määrime analoogpesasse 0 ühendatud
temperatuurianduri sisendiks
// digitaalpesade väljundiks määramine
pinMode(pinA, OUTPUT);
pinMode(pinB, OUTPUT);
pinMode(pinC, OUTPUT);
pinMode(pinD, OUTPUT);
pinMode(pinE, OUTPUT);
pinMode(pinF, OUTPUT);
pinMode(pinG, OUTPUT);
pinMode(D1, OUTPUT);
pinMode(D2, OUTPUT);
pinMode(D3, OUTPUT);
pinMode(D4, OUTPUT);
}

void loop() {
  int temperatuuriandmed = analogRead(0); //loeme temperatuuri andmeid
  float millivoldid = (temperatuuriandmed / 1024.0) * 5000;
//teisendame millivoltideks
  int temp_celsius = millivoldid / 10; //teisendame celsiuseks ja
teeme täisarvuks

  int esimene = temp_celsius /10 ; //leiame arvu esimese numbri
  Serial.println(temp_celsius); //prindime serial monitori
täisarvulise temperatuuri
  byte num1 = temp_celsius /10; //viime sobivasse tyypi

  if(esimene <0){
    byte num1 = (esimene)*(-1); //juhul kui temp on alla nulli
  }
  Serial.println(num1); //prindime Serial monitori esimene numbri
  byte num2 = temp_celsius % 10; //leiame teise numbri
  Serial.println(num2); //prindime ka selle Serial Monitori

  algus = millis(); //algusaeg
  lopp = algus; //väärtustame lõpuni jääva aja algusajaga
  //tsükkel näidiku uuendamiseks 1 sekundi tagant, vastasel juhul
  uuendatakse näidikut liiga kiirest ja vaadata on ebameeldiv
  while ((lopp - algus) <= 1000){ //tsükkel jookseb kuni 1000
  millisekundit saab täis (kui lõpu ja algusaja vahe on 1000 ms)
  //esimene koht
  digitalWrite(D1, HIGH);
  digitalWrite(D2, LOW);
  digitalWrite(D3, LOW);
  digitalWrite(D4, LOW);
  //kui on miinuskraadid, siis kuvame sinna miinuse
  if (temp_celsius <0){
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, HIGH);
    digitalWrite(pinF, HIGH);

```

```

digitalWrite(pinG, LOW);
delay(1);
}
//näidiku teine kakarter
digitalWrite(D1, LOW);
digitalWrite(D2, HIGH);
digitalWrite(D3, LOW);
digitalWrite(D4, LOW);
//kirjutame sinna arvu esimese numbri
kirjutaNumber(num1);
delay(1);
//kolmas koht näidikul
digitalWrite(D1, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, HIGH);
digitalWrite(D4, LOW);
//kirjutame arvu teise numbri
kirjutaNumber(num2);
delay(1);
//neljas koht näidikul
digitalWrite(D1, LOW);
digitalWrite(D2, LOW);
digitalWrite(D3, LOW);
digitalWrite(D4, HIGH);
//joonistame temperatuurimärgi
digitalWrite(pinA, LOW);
digitalWrite(pinB, LOW);
digitalWrite(pinC, HIGH);
digitalWrite(pinD, HIGH);
digitalWrite(pinE, HIGH);
digitalWrite(pinF, LOW);
digitalWrite(pinG, LOW);
delay(1);
lopp = millis(); //möödame aega ja kirjutame muutujasse lopp
}
}
//funktsioon, kus defineeritakse numbrid 0-9
void kirjutaNumber(byte arv) {
  if (arv == 0){
    digitalWrite(pinA, LOW);
    digitalWrite(pinB, LOW);
    digitalWrite(pinC, LOW);
    digitalWrite(pinD, LOW);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, HIGH);
  }
  else if(arv == 1){
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, LOW);
    digitalWrite(pinC, LOW);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, HIGH);
    digitalWrite(pinF, HIGH);
  }
}

```

```

    digitalWrite(pinG, HIGH);
}
else if(arv == 2){
    digitalWrite(pinA, LOW);
    digitalWrite(pinB, LOW);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, LOW);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, HIGH);
    digitalWrite(pinG, LOW);
}
else if(arv == 3){
    digitalWrite(pinA, LOW);
    digitalWrite(pinB, LOW);
    digitalWrite(pinC, LOW);
    digitalWrite(pinD, LOW);
    digitalWrite(pinE, HIGH);
    digitalWrite(pinF, HIGH);
    digitalWrite(pinG, LOW);
}
else if(arv == 4){
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, LOW);
    digitalWrite(pinC, LOW);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, HIGH);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, LOW);
}
else if(arv == 5){
    digitalWrite(pinA, LOW);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, LOW);
    digitalWrite(pinD, LOW);
    digitalWrite(pinE, HIGH);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, LOW);
}
else if(arv == 6){
    digitalWrite(pinA, LOW);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, LOW);
    digitalWrite(pinD, LOW);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, LOW);
}
else if(arv == 7){
    digitalWrite(pinA, LOW);
    digitalWrite(pinB, LOW);
    digitalWrite(pinC, LOW);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, HIGH);
    digitalWrite(pinF, HIGH);
}

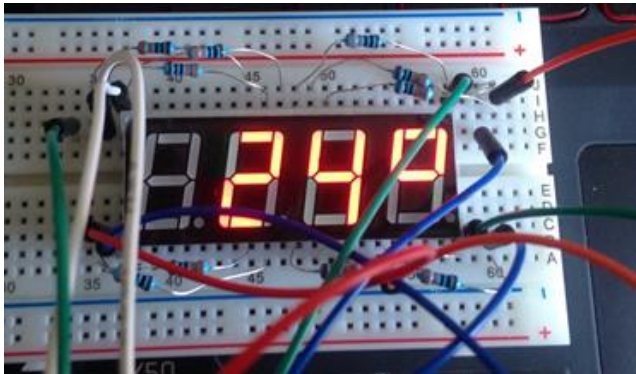
```

```

    digitalWrite(pinG, HIGH);
}
else if(arv == 8){
    digitalWrite(pinA, LOW);
    digitalWrite(pinB, LOW);
    digitalWrite(pinC, LOW);
    digitalWrite(pinD, LOW);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, LOW);
}
else if(arv == 9){
    digitalWrite(pinA, LOW);
    digitalWrite(pinB, LOW);
    digitalWrite(pinC, LOW);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, HIGH);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, LOW);
}
}
}

```

Joonis 91. Kood termomeetri koostamiseks [67]



Joonis 92. Termomeeter

Käesolevas projektis kuvatakse ainult numbreid, lisaks võib proovida punktide sisse lülitamist ning temperatuuri kuvamist ühe komakoha täpsusega. Samuti võib mõõtmistulemust täpsemaks muuta, lisades õiged ümardamisreeglid. Lisaks saab numbrite 0-9 defineerimiseks kasutada vähem ruumi võtvat maatriksina defineerimise moodust ning if-else laused asendada switch-case lausetega. Lisalahendusena saab näidikul kuvada ka näiteks kellaaja mooduliga mõõdetavat kellaega.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli tutvustada Arduinoga kasutatavaid komponente ja andureid ning koostada neist huvitavamate kohta eestikeelsed õppematerjalid. Töös anti ülevaade Arduino platvormist, kirjeldati stardikomplektis sisalduvaid komponente ning toodi näiteid nende kasutusvõimaluste kohta. Koostati juhend arenduskeskkonna üles seadmiseks ning umbes kümne tunni jagu kasvava raskusastmega õppematerjale, mis annavad ülevaate huvitavamate andurite ja lisakomponentide praktilisest kasutamisest ning programmeerimise algtõdedest.

Töö tulemusena valminud õppematerjalid on sobilikud kasutamiseks robotikaringides, reaalinetaundides ning ka iseseisvaks robotika õppimiseks. Materjalid on sobilikud alates põhikoolitasemest, kuid projektide lõpus olevate edasiarendusideede tõttu pakuvad piisavalt väljakutset ka gümnaasistidele.

Materjalid annavad ülevaate Arduino platvormi ning peamiste andurite kasutusvõimalustest ning tagavad piisavad alusteadmised, et edasi suunduda näiteks robotite ehitamise või targa maja projektide koostamise juurde.

Bakalaureusetöö kirjutamise käigus tutvus autor lähemalt Arduino platvormiga, sai praktilist kogemust riistvaralähedase programmeerimise valdkonnas ning õppematerjalide koostamise kogemuse.

Kasutatud kirjandus

- [1] K. K. Eno Tõnisson, „Informaatikaõpe koolis – kas ja kuidas õpetada?“, Õpetajate leht, 05. 06. 2015. [Võrgumaterjal]. Available: <http://opleht.ee/23475-informaatikaope-koolis-kas-ja-kuidas-opetada/>. [Kasutatud 12. 05. 2016].
- [2] „What is Arduino?“, Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Guide/Introduction>. [Kasutatud 14. 12. 2015].
- [3] „Arduino Products“, Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Guide/Introduction>. [Kasutatud 14. 12. 2015].
- [4] D. Kushner, „The Making of Arduino“, IEEE, 26 10 2011. [Võrgumaterjal]. Available: <http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>. [Kasutatud 28. 01. 2016].
- [5] „Cover“, Wiring, [Võrgumaterjal]. Available: <http://wiring.org.co/>. [Kasutatud 28. 04. 2016].
- [6] „Frequently Asked Questions“, Arduino. [Võrgumaterjal]. [Kasutatud 01. 02. 2016].
- [7] J. Provost, „Why the Arduino Won and Why It's Here to Stay“, Make, 2011. [Võrgumaterjal]. Available: <http://makezine.com/2011/02/10/why-the-arduino-won-and-why-its-here-to-stay/>. [Kasutatud 02. 02. 2016].
- [8] „Google Trends“, Google, [Võrgumaterjal]. Available: <https://www.google.com/trends/explore#q=arduino%2C%20raspberry%20pi%2C%20LEGO%20mindstorms&cmpt=q&tz=Etc%2FGMT-2>. [Kasutatud 03. 02. 2016].
- [9] „Arduino Wars: Genuino vs the Phantom Arduino and the Clones“, Make, [Võrgumaterjal]. Available: <http://makezine.com/2015/06/22/arduino-wars-genuino-vs-phantom-arduino-clones/>. [Kasutatud 12. 05. 2016].
- [10] „Genuino Brand“, Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/GenuinoBrand>. [Kasutatud 23 02 2016].
- [11] „Arduino UNO & Genuino UNO“, Arduino. [Võrgumaterjal]. [Kasutatud 23 02 2016].
- [12] M. Banzi, „Send in the clones“, Arduino, 10 07 2013. [Võrgumaterjal]. Available: <https://blog.arduino.cc/2013/07/10/send-in-the-clones/>. [Kasutatud 02. 02. 2016].
- [13] „How To Spot A Counterfeit, Cloned or Fake Arduino“, Codeduino, [Võrgumaterjal]. Available: <https://codeduino.com/tutorials/how-to-spot-a-fake-arduino/>. [Kasutatud 02. 02. 2016].
- [14] „Arduino Uno Rev3-Main Board“, Cytron Technologies, [Võrgumaterjal]. Available: <http://www.cytron.com.my/p-arduino-uno>. [Kasutatud 20. 02. 2016].
- [15] „Arduino 101 & Genuino 101“, Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoBoard101>. [Kasutatud 20. 02. 2016].
- [16] J. Condliffe, „This New Arduino 101 Uses Intel's Tiny, Low-Power Curie Chip“, Gizmodo, 16 10 2015. [Võrgumaterjal]. Available: <http://gizmodo.com/this-new-arduino-101-uses-intels-tiny-low-power-curie-1736895382>. [Kasutatud 20. 02. 2016].

- [17] „Arduino Pro,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardPro>. [Kasutatud 20. 02. 2016].
- [18] „Arduino Pro Mini,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardProMini>. [Kasutatud 21. 02. 2016].
- [19] „Arduino MICRO & Genuino MICRO,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardMicro>. [Kasutatud 21. 02. 2016].
- [20] „Low-cost embedded development kits accelerate design cycles,“ Embedded Computing Design, [Võrgumaterjal]. Available: <http://embedded-computing.com/articles/low-cost-embedded-development-kits-accelerate-design-cycles/>. [Kasutatud 21. 02. 2016].
- [21] „Arduino Nano,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardNano>. [Kasutatud 21. 02. 2016].
- [22] „Arduino MEGA 2560 & Genuino MEGA 2560,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>. [Kasutatud 21. 02. 2016].
- [23] „Arduino Mega 2560,“ Riecktron, [Võrgumaterjal]. Available: <https://www.riektron.co.za/en/product/211>. [Kasutatud 21. 02. 2016].
- [24] „Arduino Zero & Genuino Zero,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardZero>. [Kasutatud 21. 02. 2016].
- [25] Z. Romano, „Meet Arduino ZERO,“ Arduino, 15 05 2014. [Võrgumaterjal]. Available: <https://blog.arduino.cc/2014/05/15/meet-arduino-zero/>. [Kasutatud 21. 02. 2016].
- [26] „Arduino Due,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardDue>. [Kasutatud 21. 02. 2016].
- [27] „Arduino Yún,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardYun>. [Kasutatud 22. 02. 2016].
- [28] „Arduino Gemma,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoGemma>. [Kasutatud 22. 02. 2016].
- [29] „LilyPad Arduino Main Board,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoBoardLilyPad>. [Kasutatud 22. 02. 2016].
- [30] „Arduino Motor Shield,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoMotorShieldR3>. [Kasutatud 22. 02. 2016].
- [31] „Arduino Ethernet Shield,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoEthernetShield>. [Kasutatud 22. 02. 2016].
- [32] „Arduino Proto Shield,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoProtoShield>. [Kasutatud 23. 02. 2016].
- [33] „Arduino GSM Shield,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoGSMShield>. [Kasutatud 23. 02. 2016].
- [34] „Arduino WiFi Shield 101,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoWiFiShield101>. [Kasutatud 22. 02. 2016].
- [35] „Arduino Starter Kit & Genuino Starter Kit,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoStarterKit>. [Kasutatud 24. 02. 2016].
- [36] „Arduino Basic Kit,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Main/ArduinoBasicKit>. [Kasutatud 24. 02. 2016].

- [37] „The best Arduino starter kits compared and reviewed,“ Pretzel Logix, 09. 10. 2014. [Võrgumaterjal]. Available: <https://www.pretzellogix.net/2014/10/09/three-arduino-starter-kits-compared-and-reviewed/>. [Kasutatud 07. 05. 2016].
- [38] „Arduino Basic Kit,“ Autodesk 123D Circuits, [Võrgumaterjal]. Available: <https://123d.circuits.io/shop/arduino>. [Kasutatud 07. 05. 2016].
- [39] „Ultimate UNO R3 Starter Kit for Arduino/Infiduiino,“ ebay, [Võrgumaterjal]. Available: <http://www.ebay.com/itm/New-Ultimate-UNO-R3-Starter-Kit-for-Arduino-Infiduiino-LCD-Big-Two-Layer-Box-/361461753021?hash=item5428cca4bd:g:4vwAAOSwvgdW46S4>. [Kasutatud 07. 05. 2016].
- [40] „Arduino,“ Wikipedia, [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Arduino#cite_note-AutoF7-38-23. [Kasutatud 01. 03. 2016].
- [41] „Using AVR Studio 5 with Arduino projects,“ EngBlaze, [Võrgumaterjal]. Available: <http://www.engblaze.com/tutorial-using-avr-studio-5-with-arduino-projects/>. [Kasutatud 01. 03. 2016].
- [42] „Arduino Software (IDE),“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Guide/Environment>. [Kasutatud 01. 03. 2016].
- [43] „Raspberry Pi 3 - Model B,“ Modmypi, [Võrgumaterjal]. Available: <http://www.modmypi.com/raspberry-pi/rpi3-model-b/raspberry-pi-3-model-b-and-8gb-microsd-card-bundle>. [Kasutatud 05. 05. 2016].
- [44] „Raspberry Pi 3 Model B,“ Raspberry Pi Foundation, [Võrgumaterjal]. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. [Kasutatud 05. 05. 2016].
- [45] „Nutiseadmest kaugjuhitava roboti ehitamine Raspberry Pi baasil,“ Arvutiteaduse instituut, [Võrgumaterjal]. Available: <https://courses.cs.ut.ee/t/kids/Main/Nutirobot>. [Kasutatud 05. 05. 2016].
- [46] R. Sell, „Valgusdiod,“ Robotika Kodulabor, [Võrgumaterjal]. Available: <http://home.roboticlab.eu/et/examples/digi/led>. [Kasutatud 07. 05. 2016].
- [47] L. Summers, „Calculating Required Resistor Values,“ 25. 05. 2014. [Võrgumaterjal]. Available: <http://www.filearchivehaven.com/2014/05/25/arduino-calculating-required-resistor-values/>. [Kasutatud 10. 05. 2016].
- [48] A. Ainla, „Roboti ehitamise juhend - Servomootor,“ 2003. [Võrgumaterjal]. Available: <http://digi.physic.ut.ee/mw/images/7/73/Servomootor.pdf>. [Kasutatud 09. 05. 2016].
- [49] A. Ainla, „Robotiehitamise juhend - Sammmootor,“ 2003. [Võrgumaterjal]. Available: <http://digi.physic.ut.ee/mypages/oppetoo/robotex/vahendid/files/Sammootor.pdf>. [Kasutatud 10. 05. 2016].
- [50] „LiquidCrystal "Hello World",“ Arduino, 17. 08. 2015. [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Tutorial/HelloWorld>. [Kasutatud 27. 04. 2016].
- [51] „Photo resistor,“ Resistor Guide, [Võrgumaterjal]. Available: <http://www.resistorguide.com/photoresistor/>. [Kasutatud 29. 04. 2016].

- [52] S. Electronics, „Photoresistor,“ SparkFun Electronics, [Võrgumaterjal]. Available: <http://www.exploringarduino.com/parts/photoresistor/>. [Kasutatud 29. 04. 2016].
- [53] P. R.V, „Interfacing Sensors - Microphone Sound Detection Sensor,“ 17. 02. 2014. [Võrgumaterjal]. Available: <http://www.induino.com/2014/02/interfacing-sensors-microphone-sound.html>. [Kasutatud 10. 05. 2016].
- [54] „How to Build a LM35 Temperature Sensor Circuit,“ LearningaboutElectronics, [Võrgumaterjal]. Available: <http://www.learningaboutelectronics.com/Articles/LM35-temperature-sensor-circuit.php>. [Kasutatud 10. 05. 2016].
- [55] N. Semiconductors, „Enabling Timekeeping Function and Prolonging Battery Life in Low Power Systems,“ Digi-Key Electronics , 14 12 2011. [Võrgumaterjal]. Available: <http://www.digkey.com/en/articles/techzone/2011/dec/enabling-timekeeping-function-and-prolonging-battery-life-in-low-power-systems>. [Kasutatud 05. 09. 2016].
- [56] „Arduino Modules - Flame Sensor,“ Instructables, [Võrgumaterjal]. Available: <http://www.instructables.com/id/Arduino-Modules-Flame-Sensor/>. [Kasutatud 05. 09. 2016].
- [57] R. Elliott, „Beginners' Guide to Potentiometers,“ 22. 01. 2002. [Võrgumaterjal]. Available: <http://sound.westhost.com/pots.htm>. [Kasutatud 11. 05. 2016].
- [58] „LedControl,“ Arduino, [Võrgumaterjal]. Available: <http://playground.arduino.cc/Main/LedControl>. [Kasutatud 10. 05. 2016].
- [59] C. M. a. T. Igoe, „Serial to Parallel Shifting-Out with a 74HC595,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Tutorial/ShiftOut>. [Kasutatud 11. 05. 2016].
- [60] „Transistor Motor Control,“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Tutorial/TransistorMotorControl>. [Kasutatud 11. 05. 2016].
- [61] „SS8050 NPN Epitaxial Silicon Transistor,“ Fairchild. [Võrgumaterjal]. Available: <http://www.mouser.com/ds/2/149/SS8050-117753.pdf>. [Kasutatud 11. 05. 2016].
- [62] „TinkerKit Relay Module,“ Arduino, [Võrgumaterjal]. Available: <https://store.arduino.cc/product/T010010>. [Kasutatud 10. 05. 2016].
- [63] „createChar(),“ Arduino, [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Reference/LiquidCrystalCreateChar>. [Kasutatud 11. 05. 2016].
- [64] „Servo Control with Joystick and Arduino,“ Explore Labs, [Võrgumaterjal]. Available: <http://learn.explorelabs.com/servo-control-with-joystick-and-arduino/>. [Kasutatud 28. 04. 2016].
- [65] T. Igoe, „Stepper Speed Control,“ Arduino, 28. 11. 2010. [Võrgumaterjal]. Available: <https://www.arduino.cc/en/Tutorial/StepperSpeedControl>. [Kasutatud 28. 04. 2016].
- [66] K. Shirriff, „Arduino-IRremote,“ 2009-2012. [Võrgumaterjal]. Available: <https://github.com/z3t0/Arduino-IRremote>. [Kasutatud 29. 04. 2016].
- [67] K. Blåsol, „Arduino and the 4 digit 7 segment led display,“ 05. 12. 2015. [Võrgumaterjal]. Available: <https://www.youtube.com/watch?v=iZl1GjCvliw>. [Kasutatud 10. 05. 2016].

- [68] „Using a Temp Sensor,“ 19. 11. 2015. [Võrgumaterjal]. Available: <https://learn.adafruit.com/tmp36-temperature-sensor/using-a-temp-sensor>. [Kasutatud 09. 05. 2016].

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Selena Lubi,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Arduino stardikomplekti õppematerjalid“, mille mille juhendajad on Anne Villemis, Taavi Duvin ja Alo Peets,

1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 12.05.2016