

TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
Arvutiteaduse instituut
Informaatika eriala

Lauri Uusmaa

**Arduino kasutamine koos LEGO Mindstorms EV3
robotiga**

Bakalaureusetöö (9 EAP)

Juhendajad: Anne Villems
Taavi Duvin
Alo Peets

TARTU 2015

Arduino kasutamine koos LEGO Mindstorms EV3 robotiga

Lühikokkuvõte:

Käesoleva bakalaureusetöö eesmärgiks on luua eestikeelne õppematerjal LEGO Mindstorms EV3 ja Arduino ühendamise ning kooskasutamisest. Töö sisu on jaotatud kolme peatükki. Esimeses peatükis tutvustatakse Arduino platvormi. Teises peatükis antakse ülevaade LEGO Mindstorms EV3 juhtploki ja Arduino arendusplaadi ühendamise kohta. Kolmandas peatükis tuuakse viis kasvava raksusastmega ülesannet koos lahenduste ja selgitustega.

Võtmesõnad: LEGO Mindstorms EV3, Arduino, robotika

Connecting the Arduino with the LEGO Mindstorms EV3

Abstract:

The aim of this bachelor thesis is to create learning material in Estonian for using the Arduino and the LEGO Mindstorms EV3 together. This thesis is divided into three chapters. The first chapter introduces Arduino platform. The second chapter gives an overview of connecting the Arduino development board and the LEGO Mindstorms EV3 programmable brick. The third chapter consists of five exercises with different difficulty levels. Exercises include solutions and explanations.

Keywords: LEGO Mindstorms EV3, Arduino, robotics

Sisukord

Sissejuhatus	5
1. Arduino ülevaade	7
1.1 Arendusplaatide populaarsus	7
1.2 Arduino võrdlus teiste tuntud platvormidega	8
1.3 Arduino tutvustus	9
1.4 Arduino olulisus prototüüpimisel	10
1.5 Erinevad Arduino arendusplaatide mudelid ja nende riistvara	11
1.6 Arduino programmeerimine	12
1.7 Arduino kasutusala	13
1.8 Arduino kasulikkus LEGO Mindstorms EV3 robotiga ühendamisel	14
2. Arduino kasutamine LEGO Mindstorms EV3 robotiga	16
2.1 Soovitused Arduino arendusplaadi valimisel	16
2.2 I ² C siini ülevaade	17
2.3 Tarkvara Arduino ja LEGO Mindstorms EV3 ühendamiseks	17
2.3.1 Arduino arenduskeskkond	18
2.3.2 EV3 tarkvara Arduino arendusplaadiga ühendamiseks	26
2.4 Riistvara Arduino ja LEGO Mindstorms EV3 ühendamiseks	29
2.5 Toiteallikad Arduino arendusplaadi jaoks	32
2.6 Arduino ja EV3 vahelise ühenduse testimine	34

3. Ülesanded.....	39
3.1 Ülesanne 1 - valgusdiodi vilgutamine	39
3.2 Ülesanne 2 - valgusdiodi heleduse muutmine EV3 mootori abil	43
3.3 Ülesanne 3 - potentsiomeetriga EV3 suure mootori kiiruse muutmine.....	45
3.4 Ülesanne 4 - LEGO klotsidest ehitatud auto esirataste pööramine servomootoriga .	48
3.5 Ülesanne 5 - puldist juhitud auto, millel on esi ja tagatuled.....	53
Kokkuvõte	58
Kasutatud kirjandus.....	59
Lisad.....	61
I. Arduino arendusplaatide mudelite riistvaraline võrdlus	61
II. Töö lisana kaasa tulevad failid	62
III. Litsents.....	63

Sissejuhatus

Tänapäeval valitseb õpilaste seas olukord, kus reaalinete seotud teadmiste omandamine osutub väga keeruliseks või kohati isegi igavaks. Põhjuseks on liigne teoreetilise materjali hulk ning väheste seoste toomine päriselust. Kuigi teooria omandamine on tähtis, pole sellest eriti kasu, kui teooriat praktiliste näidetega ei seostata. Praktilised oskused aga aitavad teadmisi paremini kinnistada ja arendavad loogilist mõtlemist.

Olukorra parandamiseks ja reaalinete populariseerimiseks on hakatud õpilastele tutvustama koolirobootikat. Koolirobootikas kasutatakse enamasti LEGO Mindstorms hariduslikke robootikakomplekte, millest kõige uuem on LEGO Mindstorms EV3 hariduslik baaskomplekt (ingl k. *LEGO Mindstorms Education EV3 core set*). LEGO Mindstorms EV3 hariduslikku baaskomplekti kuuluvast programmeeritavast juhtploki, sensoritest, mootoritest ja LEGO klotsidest saab ehitada erinevaid roboteid. Neid roboteid kasutatakse, et muuta õppimine huvitavaks ja praktiliseks.

LEGO Mindstorms EV3 tuli turule 1. septembril 2013. Paari aastaga on loodud eestikeelseid õppematerjale nii EV3 juhtploki, kui ka juhtploki ühilduvate sensorite kasutamiseks. Sellegi poolest on eestikeelsetest õppematerjalidest puudus. Veelgi enam, saadavalolevad õppematerjalid on ennekõike koostatud põhikooliõpilastele ja noorematele. Nii mõnedki vanemad õpilased on olemasoleva materjali omandanud ning soovivad robootikaga edasijõudnud tasemel tegeleda.

Käesoleva bakalaureuse töö eesmärgiks on luua eestikeelne õppematerjal LEGO Mindstorms EV3 juhtploki ja Arduino arendusplaadi ühendamiseks ja kooskasutamiseks. Arduino arendusplaadi ühendamine EV3 juhtploki võimaldab kasutada paljusid EV3 juhtploki mitteühilduvaid sensoreid ja muid seadmeid. Õppematerjal on vajalik, sest LEGO Mindstorms on suhteliselt uus ning kvaliteetsed eestikeelsed juhendid eelnimetatud kahe seadme ühendamiseks puuduvad. Samuti võimaldab valminud õppematerjal sujuvat üleminekut LEGO Mindstorms juhtploki programmeerimisel kasutatavalt visuaalselt programmeerimiskeelelt Arduino programmeerimiseks kasutatavale objektorienteeritud programmeerimiskeelele. Õppematerjalis tuuakse ka erineva raskusastmega ülesanded koos lahendustega EV3 ja Arduino kooskasutamise tundma õppimiseks ja praktilise kogemuse omandamiseks. Käesolevas bakalaureusetöös käsitletakse ennekõike Arduino kasutamisest ning EV3 ja Arduino vahelise

ühenduse kasutamisest tulenevaid probleeme. Õppematerjali kasutajalt eeldatakse LEGO Mindstorms EV3 kasutamise oskust algtasemel.

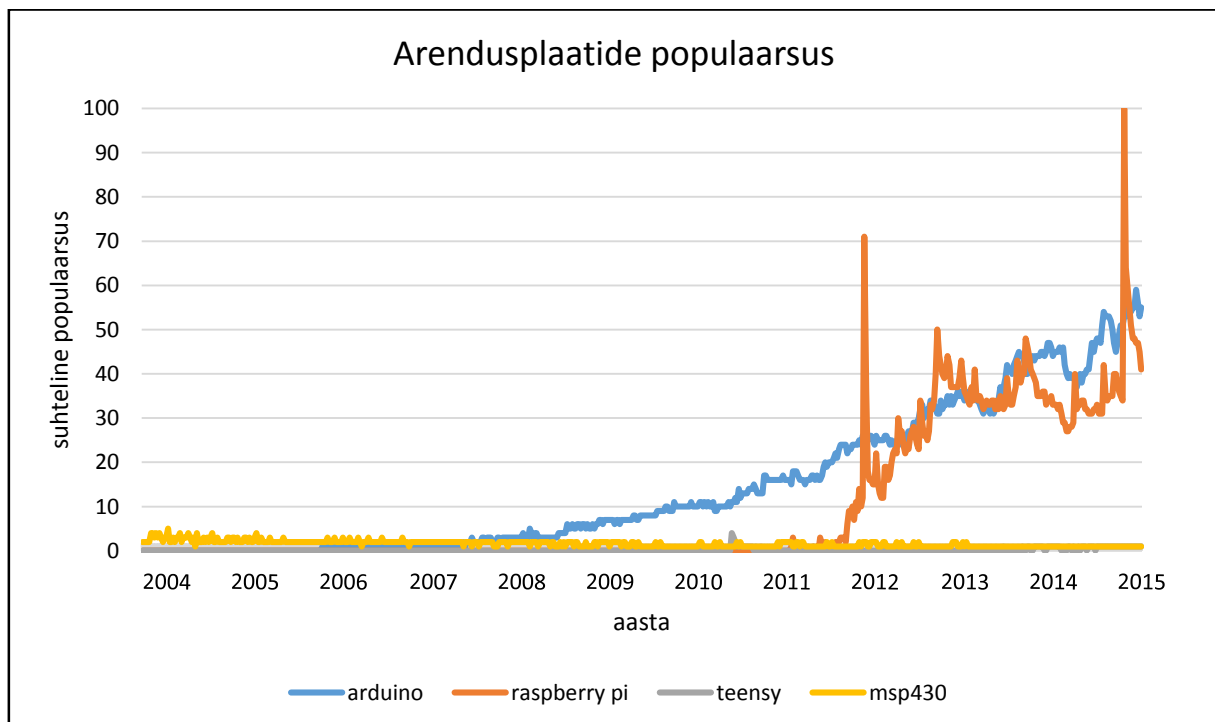
Antud bakalaureusetöö koosneb kolmest peatükist. Esimeses peatükis tutvustatakse Arduino platvormi. Teises peatükis antakse juhised Arduino arendusplaadi ja LEGO Mindstorms EV3 juhtploki ühendamiseks ning ühenduse testimiseks. Kolmandas peatükis tuuakse viis suureneva raskusastmega ülesannet EV3 ja Arduino kooskasutamiseks. Iga ülesande juures on välja toodud ülesande lahendamiseks vajalike vahendite loetelu, lahenduse idee ning näidislahendus.

1. Arduino ülevaade

Käesoleva peatüki alguses võrreldakse erinevaid kiirarenduse platvorme ja tuuakse välja miks antud bakalaureusetöös just Arduino platvormi kasutatakse. Edasi tutvustatakse Arduino platvormi, erinevaid Arduino arendusplaatide ja kirjutatakse Arduino olulisusest prototüüpimismaastikul. Samuti antakse ülevaade Arduino programmeerimisest. Peatüki lõpus tuuakse näiteid Arduino kasutusalaadest ning põhjendatakse, miks on Arduino kasulik LEGO Mindstorms EV3 robotiga ühendada.

1.1 Arendusplaatide populaarsus

Arendusplaatide valik kasvab märgatavalt iga aastaga. Nii muutub ka erinevate arendusplaatide populaarsus. Üks võimalus arendusplaatide tuntuse illustreerimiseks on kasutada avalikku veebiteenust Google Trends (<http://www.google.com/trends/>). Google Trends võimaldab sisestatud märksõnade põhjal kuvada, kui tihti üksteise suhtes on neid sõnu Google otsinguga otsitud. Joonisel 1 on kujutatud nelja tuntud arendusplaatide populaarsus.



Joonis 1. Tuntud arendusplaatide suhteline populaarsus [1].

Jooniselt 1 on näha, et Arduino populaarsus kasvab stabiilselt ning on praegu väga heal tasemel. See on ka põhjus, miks antud bakalaureusetöös kasutatakse just Arduino platvormi LEGO Mindstorms EV3 juhtplokiga ühendamiseks.

Järgnevalt võrreldakse Arduino platvormi teiste tuntud kiirarenduse platvormidega.

1.2 Arduino võrdlus teiste tuntud platvormidega

Tuntud kiirarendusplaatide riistvaraline võrdlus on välja toodud tabelis 1.

Tabel 1. Tuntud arendusplaatide võrdlus [2] [3] [4] [5].

Arendusplaadi nimi	Arduino Leonardo	Teensy 3.1	MSP430 Launchpad	Raspberry Pi 2 Model B
Mõõtmed (mm)	68,6 x 53,3 x 4,0	35,6 x 17,8 x 4,0	60 x 50 x 10	85,6 x 56,5 x 17
Protsessor	ATmega32u4	MK20DX256VLH7	MSP430G2553	ARM Cortex-A7
Töösagedus	16 Mhz	72 Mhz	16 MHz	900MHz
Välkmälu	32 KB	256 KB	16 KB	MicroSD kaart
Muutmälu	2,5 KB	64 KB	512 B	1 GB
EEPROM	1 KB	2 KB	0	0
Digitaal sisend-väljund viikused	20	34	14	40
Analoog sisendviikused	12	21	8	0
PWM kanaleid	7	12	8	
UART, I ² C, SPI	1,1,1	3,2,1	1,1,1	1,1,1
Tööpinge	5 V	5 V	3,3 V	5 V
Toitepinge (soovituslik)	7V-12V	5 V (USB)	5 V (USB)	5 V (USB)
Hind	21\$	19,8\$	9,99\$	35\$

Tabelist 1 on näha, et Teensy 3.1 ja MSP430 Launchpad on Arduinoga küllaltki sarnased. Raspberry Pi 2 aga erineb riistvara poolest oluliselt. See on miniarvuti, millel on tunduvalt

rohkem arvutusvõimsust ja mälu kui Arduinol. Raspberry Pi 2 peal jookseb Linuxil põhinev operatsioonisüsteem või Windows 10 [5].

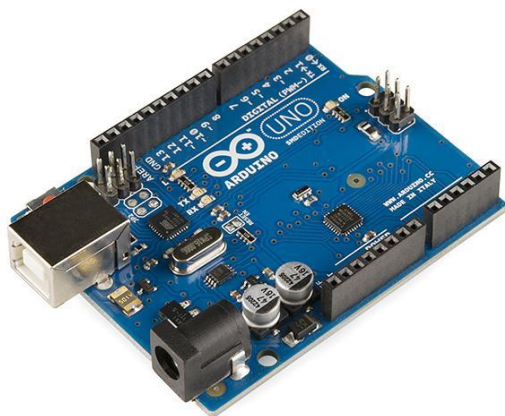
Arduino eesmärk ei ole pakkuda suurt arvutusvõimsust. Eesmärgiks on võimaldada luua roboteid, mis suudaks füüsilist maailma tajuda ja kontrollida. Arduino eeliseks teiste võrreldavate platvormidega on suur kasutajate kogukond ning olemasolevate juhendite hulk, mis muudab Arduino kasutamise lihtsaks. Just sellepärast on Arduino suurt populaarsust kogunud ning saanud enim kasutatavaks robotika platvormiks [6].

1.3 Arduino tutvustus

Käesolevas punktis on kasutatud materjale Arduino kodulehelt [7] ja Vikipeediast [8].

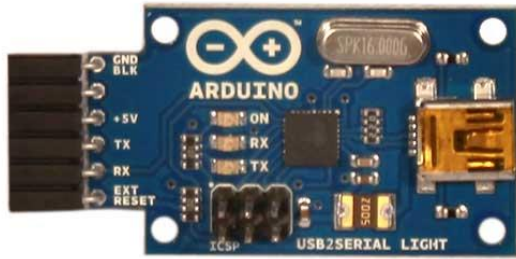
Arduino on avatud lähtekoodiga elektroonika prototüüpimiseks loodud platvorm, mis tugineb lihtsasti kasutataval riistvaral ja tarkvaral. Riistvara keskmeks on lihtne trükkplaadil asetsev mikrokontroller. Arduino tarkvara on arenduskeskkond, mille abil saab arendusplaadi jaoks programme koostada.

Arduino arendusplaat (Joonis 2) koosneb lihtsa disainiga trükkplaadist, milles kasutatakse erinevaid Atmel AVR mikrokontrollereid. Lisaks on plaadil hulk analoog ja digitaal sisend-väljund viikusid. Sisend- ja väljundviikude abil on võimalik arendusplaadiga ühendada erinevaid laiendusplaatte, sensoreid, mootoreid, tulesid jne.



Joonis 2. Arduino Uno rev3 [9].

Enamus Arduino arendusplaatide mudelid on varustatud USB liidesega, mille abil on võimalik arendusplaati ühendada arvutiga. USB ühenduse kaudu saab arendusplaadi mällu laadida programme ning kommunikatsiooni pordi (ingl k. *COM port*) kaudu saata ja vastu võtta andmeid. Need arendusplaatide mudelid, millel puudub USB liides, vajavad programmeerimiseks USB-jadaliides adapterit (Joonis 3).



Joonis 3. USB-jadaliides adapter nende Arduino arendusplaatide programmeerimiseks, millel puudub USB liides [10].

Toite saab Arduino arendusplaat USB liidese kaudu või eraldi sisendist. Toitesisendi kasutamisel sobib vastavalt mudelile 2,7V-12V alalispinge. Lisaks võib Arduino arendusplaat toitega varustada selleks ette nähtud viikude kaudu. Erinevate mudelite täpsed andmed on välja toodud lisa 1. Arduino on avatud lähtekoodiga, mis tähendab, et igaühel on võimalik Arduino kodulehelt alla laadida joonised Arduino arendusplaatidest. Nende jooniste põhjal saab vastavatest komponentidest arendusplaadi kokku panna.

1.4 Arduino olulisus prototüüpimisel

Arduino kasutamine on lihtne. Tarkvara kirjutamiseks ei pea teadma mikroprotsessori sisearhitektuuri. Paigaldades arvutisse Arduino arenduskeskkonna, saab kohe hakata kirjutama klassikalist objektorienteeritud koodi, teadmata riistvara üksikasju. See ning Arduino avatud lähtekood on ka põhjus, miks Arduino arendusplaadid on prototüüpimismaastikku oluliselt muutnud. Näiteks saab igaüks tasuta alla laadida Arduino arenduskeskkonna, kirjutada oma esimese valgusdiodi vilgutamise programmi ja selle tööd kohe arendusplaadi abil näha ning selleks kulub alla tunni aja. Eelnevalt kulus seesuguse projekti jaoks kordades rohkem aega kuna riistvara oli vaja otse programmeerida.

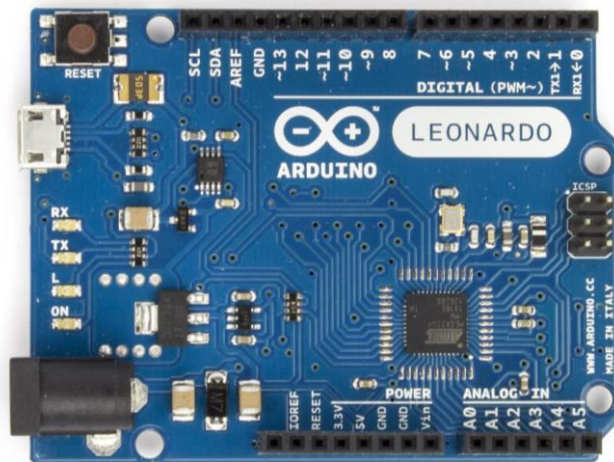
Arduino arendusplaadid ja arenduskeskkond on igapäevale lihtsasti kättesaadavad. Lisaks on Arduino programmeerimine kiiresti õpitav. Tänu sellele on lihtsamaid projekte ette võtnud ka need inimesed, kes robootikast või programmeerimisest eriti palju ei tea.

Järgnevalt võrreldakse erinevaid Arduino arendusplaatide mudeleid.

1.5 Erinevad Arduino arendusplaatide mudelid ja nende riistvara

Arduino arendusplaatidest on palju versioone, mis erinevad mõõtmetelt ja riistvara poolest [11]. Enamus mudeleid sisaldavad viievoldist lineaarset pingestabilisaatorit, mis tagab selle, et säiliks ühtlane pinge ja 16 MHz sagedusega kristall ostsillaatorit, mis varustab arendusplaati ühtlase taktisignaali.

Kõige laialdasemat kasutust on leidnud Arduino Uno (Joonis 1) ja Arduino Leonardo (Joonis 4).



Joonis 4. Arduino Leonardo [12].

Arduino Uno on vanem ja erineb Arduino Leonardost selle poolest, et USB ühenduse loomiseks kasutatakse eraldi mikrokontrollerit. [13]. Arduino Leonardo on uuem ning võrreldes Arduino Unoga on sellel vaid üks mikrokontroller ja rohkem sisend-väljund viikuseid [2]. Veel võib välja tuua Arduino Mega, LilyPad Arduino ja Arduino Yun arendusplaadid. Arduino Mega on varustatud 54 digitaalse sisend-väljund viiguga ning 16 analoog sisendviiguga, samas kui Arduino Leonardo vastavad viikude arvud on 14 ja 6 [14]. Seega on Arduino Mega sobiv projektideks, kus läheb tavapärasest rohkem viikuseid vaja. LilyPad Arduino on aga väike ja

ringjas ning on loodud riitele kinnitamiseks [15]. Arduino Yun erineb teistest selle poolest, et arendusplaadil on Ethernet ja WiFi tugi ning micro-SD kaardi pesa. Sisseehitatud WiFi, mis toetab Linuxi distributsiooni OpenWrt-Yun, teeb arendusplaadist Arduino Yun võrguühendusega arvuti [16].

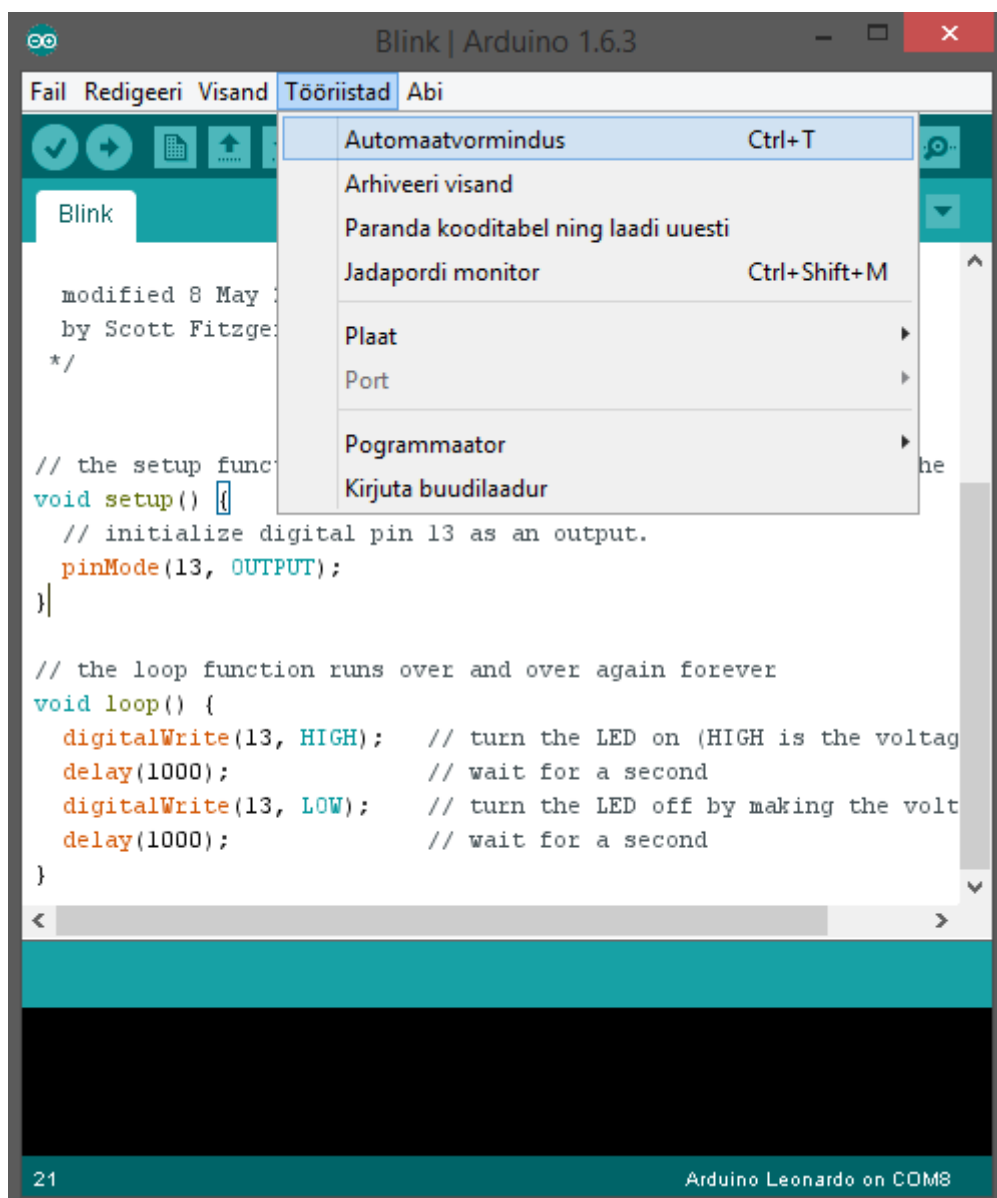
Olenemata mudelist, toimub Arduino arendusplaadi programmeerimine üldjoones ühtmoodi. Järgnevalt tutvutaksegi Arduino programmeerimisega.

1.6 Arduino programmeerimine

Arduino arendusplaadi programmeerimiseks kasutatakse Arduino IDE (Integrated Development Environment) ehk Arduino arenduskeskkonda. Arduino arenduskeskkond on Javas kirjutatud rakendus. Java tagab selle, et rakendus töötab ühtmoodi nii Windowsi, Linuxi kui ka OS X keskkonnas. Arduino arenduskeskkond lähtub projektist Wiring ja programmeerimiskeskonnast Processing [7]. Wiring on Arduinoga sarnane avatud lähtekoodiga elektroonika prototüüpimise platvorm. Wiring põhineb Processing programmeerimiskeelel. Processing on multimeedia programmeerimise keskkond ja ühtlasi ka programmeerimiskeel ning seda hakati arendama 2001. aastal eesmärgiga õpetada programmeerimise aluseid multimeedia vallas.

Arduino programm kirjutatakse C++ keele lihtsustatud variandis. Töötava programmi jaoks on vaja defineerida ainult kaks funktsiooni: *setup()* ja *loop()*. Funktsioon *setup()* käivitub programmi töö alguses ühe korra. Seda kasutatakse muutujatele algväärtuste omistamiseks ja sisend-väljund viikude seadistamiseks. Funktsioon *loop()* töötab tsüklis ning selles olev kood juhibki aktiivselt Arduino arendusplaadi tööd. Arduino arenduskeskkond on disainitud selliselt, et algajatel seda lihtne kasutada oleks, kuid võimaldab sellegipoolest programmeerimises edasijõudnutel oma oskusi rakendada.

Arenduskeskkond sisaldab koodiredaktorit, mida iseloomustavad süntaksi esiletoomine, sulgude kokkuviiimine ja koodi automaatvormindus (Joonis 5). Koodi kompileerimine ja selle üleslaadimine arendusplaadi mällu toimuvad ühe nupuvajutusega.



Joonis 5. Arduino arenduskeskkonna koodi automaatvormindus. Kursori juures oleva sulu jaoks on näidatud sulu alguskoht.

Käesolevas punktis tutvuti Arduino arendusplaadi programmeerimiseks vajamineva Arduino arenduskeskkonnaga. Edasi kirjutatakse Arduino kasutuseladest.

1.7 Arduino kasutuselad

Arduino arendusplaate kasutatakse põhiliselt robootika kiireks prototüüpimiseks ja erinevate robotite ehitamiseks. Arduino külge saab ühendada erinevaid digitaal ja analoog sensoreid, mille abil on võimalik ümbritsevast keskkonnast andmeid hankida. Näiteks on Arduinoga võimalik juhtida valgusdioode. Kui jõulupuul on sinised, punased ja valged tuled ning on vaja

neid vilkuma panna nii, et alguses põlevad sinised tuled kaks sekundit, seejärel valged tuled üks sekund ning lõpuks kõik tuled korraga neli sekundit, siis Arduinoga saab seda lihtsasti teha. Teisest küljest, kui Arduinoga ühendada temperatuuri sensor, saab temperatuuri muutumisel näiteks relee abil toa kütte sisse-välja lülitada. Ühendades Arduino arendusplaadiga valgussensori ja paigaldades selle akna juurde on võimalik toa valgustuse tugevust vastavalt tупpa jõudvale valgusele reguleerida.

Kasutades erinevaid teeke (ingl k. *library*), saab lihtsasti programmeerida väikeste vedelkristallekraanide tööd, kasutada WiFi mooduleid, suhelda erinevate mikrokontrolleritega ja palju muud. Arduinot on võimalik kasutada isegi drooni juhtimisel navigatsiooniseadmena, selleks on vaja Arduinoga ühendada GPS moodul.

Arduino ei sobi erilahendusteks nagu näiteks kohad kus on vaja suurt stabiilsust või riistavara kõrget efektiivsust. Küll aga sobib Arduino väga hästi kiirprototüüpimiseks ja projektideks, kus stabiilsus pole kõige tähtsam. Arduino arendusplaatide jaoks on kasutusalasid palju, piiranguks on vaid Arduino jõudlus ning mälu maht.

Antud punktis toodi näiteid Arduino kasutusaladest. Järgnevalt leitakse põhjusi Arduino ühendamiseks LEGO Mindstorms EV3 robotiga.

1.8 Arduino kasulikkus LEGO Mindstorms EV3 robotiga ühendamisel

Arduino ühendamine LEGO Mindstorms EV3 robotiga pakub uusi võimalusi nii sensorite ühendamiseks kui ka robotikakomplekti programmeerimiseks. Lego Mindstorms EV3 sensorid on kõik digitaalsed ning analoogsensoreid otse EV3 külge ühendada pole võimalik. Arduino arendusplaat on aga nagu vahelüli, mis suudab ümbritsevast keskkonnast nii analoog- kui ka digitaalsensorite abil erinevaid andmeid koguda ja need EV3 robotile saata. Samuti saab EV3 robotist saata andmeid Arduinole, mis siis saadud andmete põhjal erinevaid väljundseadmeid juhtida võimaldab.

LEGO Mindstorms EV3 robotil on neli sisendporti ja neli väljundporti. Seega on samal ajal kasutatavate sisend- ja väljundseadmete arv piiratud. Arduino arendusplaat aga ühendatakse EV3 robotiga vaid ühe pordi abil. Arduino arendusplaadi külge saab ühendada mitmeid

sensoreid ning erinevaid väljundseadmeid. Seega kasutades Arduino arendusplaati saab ühe pordi kaudu ühendada mitu seadet.

Arduino arendusplaatidega ühilduvad väga paljud sensorid ja muud seadmed. Arduino sensorid ei vaja spetsiaalset liidest või andmeedastusprotokolli. LEGO Mindstorms sensorid seevastu on aga terviklikud plastikust korpuse sees olevad seadmed, mille andmevahetus käib spetsiaalselt LEGO Mindstorms jaoks loodud liidese kaudu. Seega võimaldab Arduino arendusplaadi ühendamine EV3 külge paremini tehnoloogiaga kaasas käia, sest Arduino arendusplaadiga ühilduvaid sensoreid ja väljundseadmeid uuendatakse tihedamini. Veelgi enam, kui luuakse mingi uus sensor, siis suure tõenäosusega luuakse ka Arduinoga ühilduv mudel sellest sensorist. LEGO Mindstorms EV3 robotile uue sensori loomine võtab aga kauem aega. Lisaks eelnevale on Arduinoga ühilduvad sensorid ja seadmed kordades odavamad kui sama funktsionaalsusega EV3 sensorid ja seadmed.

LEGO Mindstorms EV3 programmeerimiseks kasutatakse visuaalset programmeerimiskeelt. Arduino programm seevastu aga kirjutatakse objektorienteeritud keeles. Arduino programmeerimine ja selleks kasutatav keel on tavakasutajale lihtsaks tehtud. See lubab EV3 programmeerimise põhimõtteid lihtsasti Arduino programmeerimisse üle kanda. Seega võimaldab Arduino ja EV3 ühendamine sujuvat üleminekut visuaalselt programmeerimiskeelelt objektorienteeritud programmeerimiskeelele. Nii on võimalik sujuvalt üle minna LEGO Mindstorms tarkvara abil programmeerimiselt objektorienteeritud keeles programmeerimisele ning hakata robotikaga tegelema juba edasijõudnud tasemel.

Järgmises peatükis kirjeldatakse kuidas ühendada Arduino arendusplaati LEGO Mindstorm EV3 robotiga.

2. Arduino kasutamine LEGO Mindstorms EV3 robotiga

Käesolevas peatükis kirjeldatakse kuidas ühendada Arduino arendusplaat LEGO Mindstorms EV3 juhtploki. Peatüki esimeses pooles antakse soovitusi sobiva Arduino arendusplaadi valimiseks, tutvustatakse I²C (Inter-Integrated Circuit) siini ning antakse juhised Arduino arendusplaadi ja EV3 juhtploki programmeerimiseks vajaminevast tarkvarast. Peatüki teises pooles kirjeldatakse kuidas riistvara poole pealt ühendada Arduino arendusplaat ja EV3 juhtplokk. Samuti kirjutatakse erinevatest võimalustest Arduino arendusplaadi toitmiseks ning antakse juhised Arduino arendusplaadi ja EV3 juhtploki vahelise ühenduse testimiseks.

Selguse mõttes tuuakse välja sõnade “juhe” ja “kaabel” tähendused käesolevas juhendis. Sõna “juhe” all mõeldakse ühesoonelist isoleerkattega painduvat elektrijuhti ning sõna “kaabel” all mõeldakse mitme isoleeritud soonega elektrijuhti, mida ümbritseb hermeetiline kest.

2.1 Soovitused Arduino arendusplaadi valimisel

Arduino arendusplaatidel on üks hea omadus. Nimelt töötab sama Arduino programm erinevate Arduino arendusplaatide mudelite peal sama moodi, kui muidugi kõrvale jätta mõne mudeli erifunktsioonid ja viikude konfiguratsioon. Seega Arduino programmeerimise poole pealt märkimisväärselt erinevust pole, missugune arendusplaat valida. Järgnevalt antaksegi soovitusi Arduino arendusplaadi valikul.

Arduino arendusplaadi valik langetatakse enamasti kahe kriteeriumi alusel - esiteks, ühilduvus erinevate laiendusplaatidega ning teiseks, arendusplaadi suurus. Arduino Uno ja Arduino Leonardo on hea valik sel juhul, kui tekib vajadus lisada mõni laiendusplaat, sest nende kahe arendusplaadiga sobivad enamus laiendusplaat. Kui oluline on aga suurus, siis on mõistlik valida Arduino Mini või Arduino Micro. Arduino Mini on eelnevast kahest väiksem (30mm x 18mm), kuid leppida tuleb USB liidese puudumisega. Arduino Mini saab seega programmeerida ainult USB-jadaliides adapteri abil, mis ühendatakse arendusplaadi vastavate viikude külge. Arduino Mini küljes on 14 digitaalviiku ja 8 analoogviiku. Kasutusmugavuse mõttes on hea valik Arduino Micro, millel on USB liides olemas. Arduino Micro sarnaneb riistvara poolest Arduino Leonardoga, kuid on mahutatud väiksema (48mm x 18mm) plaadi peale. Kui esmatähtis on aga arendusplaadi viikude arv ja mälumaht, siis on suurepärase valik

Arduino Mega, millel on 54 digitaalviiku ja 16 analoogviiku. Lisaks on Arduino Mega välmälu maht 128KB ning SRAM maht 4KB.

2.2 I²C siini ülevaade

Arduino ja EV3 vaheliseks suhtluseks kasutatakse I²C siini. Järgnevalt on toodud lühike ülevaade I²C siini tööpõhimõttest.

I²C (Inter-Integrated Circuit) on kahesuunaline kahesooneline jadasiin. Seda nimetatakse ka “kahe juhtme liideseks”, kuna suhtluseks kasutatakse kahte juhet. Üks juhe kannab taktsignaali, mis tagab selle, et ühenduses olevad seadmed saadavad andmeid samal kiirusel. Sellele juhtmele viidatakse kui “SCL” (Serial Clock Line). Teine juhe on andmete saatmiseks, kusjuures andmeid saadetakse ja võetakse vastu samal ajal. Sellele juhtmele viidatakse kui “SDA” (Serial Data Line).

I²C seadistuses ühendatud seadmeid nimetatakse ülemaks ja alluvaks. Ülemseade kontrollib alati suhtlust ja seab suhtluse taktsageduse mõlema seadme jaoks. Seadet, mida juhitakse, nimetatakse alamseadmeks. Kui ülemseade saadab käsu, siis alamseade lõpetab käesoleva tegevuse ja kuulab. Kui ülemseade küsib andmeid, siis alamseade saadab ülemseadmele need andmed.

Kirjeldatai I²C siini tööpõhimõtet. Järgnevalt antakse ülevaade Arduino arendusplaadi ja EV3 juhtploki omavahelise ühendamise jaoks vajaminevast tarkvarast.

2.3 Tarkvara Arduino ja LEGO Mindstorms EV3 ühendamiseks

Tarkvara jaguneb kaheks osaks. Üks osa on Arduino IDE, milles luuakse programm Arduino arendusplaadi jaoks. Teine osa on LEGO Mindstorms EV3 tarkvara, milles luuakse programm EV3 jaoks. EV3 programm saadab andmeid Arduinole ja/või võtab andmeid Arduinolt vastu.

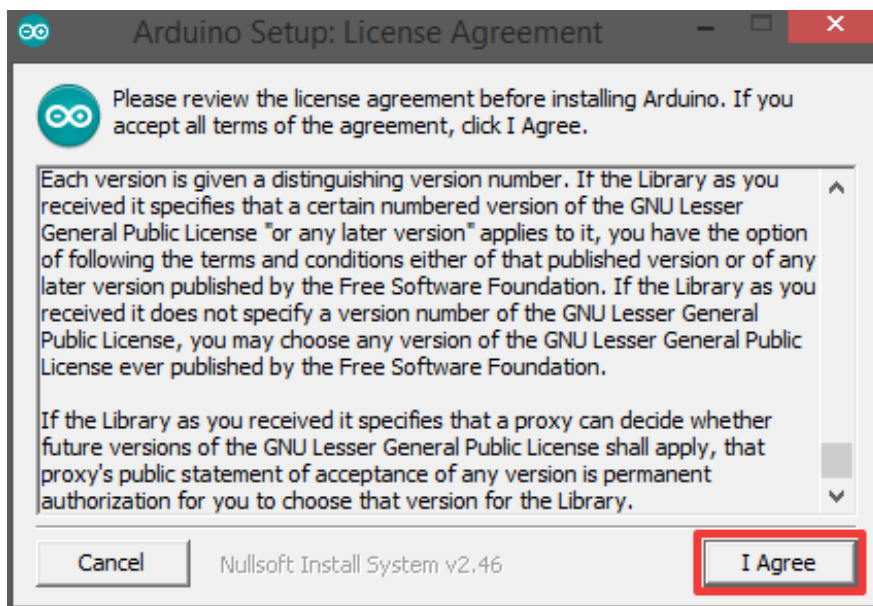
Järgmises kahes punktis on toodud juhised vajaliku tarkvara paigalduseks. Antud õpetus on operatsioonisüsteemi Windows jaoks.

2.3.1 Arduino arenduskeskkond

Arduino arendusplaadi programmeerimiseks läheb vaja Arduino IDE ehk Arduino arenduskeskkonda. Arduino IDE saab alla laadida Arduino kodulehelt “Download” sektsioonist (<http://www.arduino.cc/en/Main/Software>). Antud juhendis on kasutatud kõige uuemat versiooni, milleks on versioon 1.6.3. Järgnevalt kirjeldatakse vajalikke samme Arduino IDE paigaldamiseks.

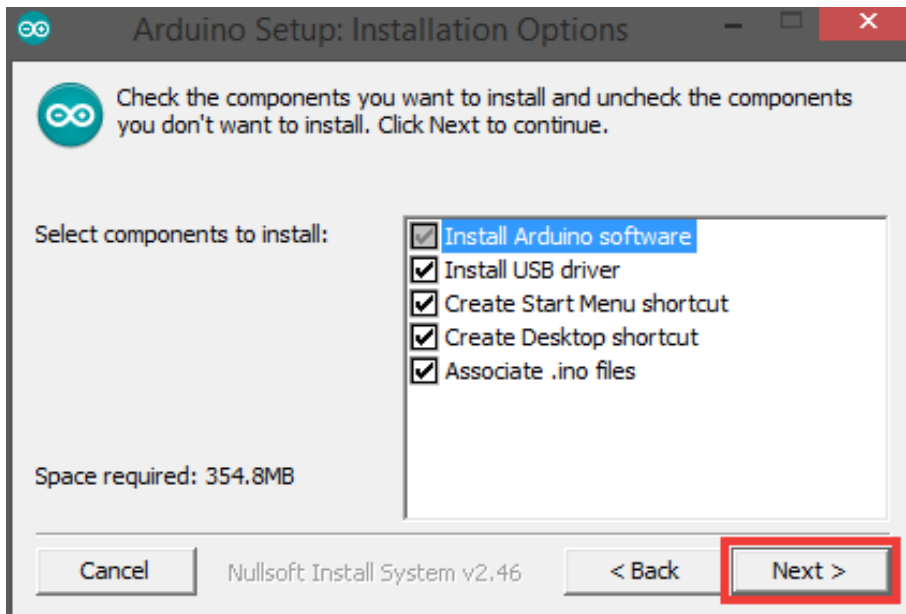
Peale faili allalaadimist ja käivitamist avaneb programmi paigaldusviisard. Programmi paigaldamiseks toimida järgmiselt:

1. Peale kasutustingimuste läbi lugemist vajutada “I Agree” nupule, et nendega nõustuda (Joonis 6).



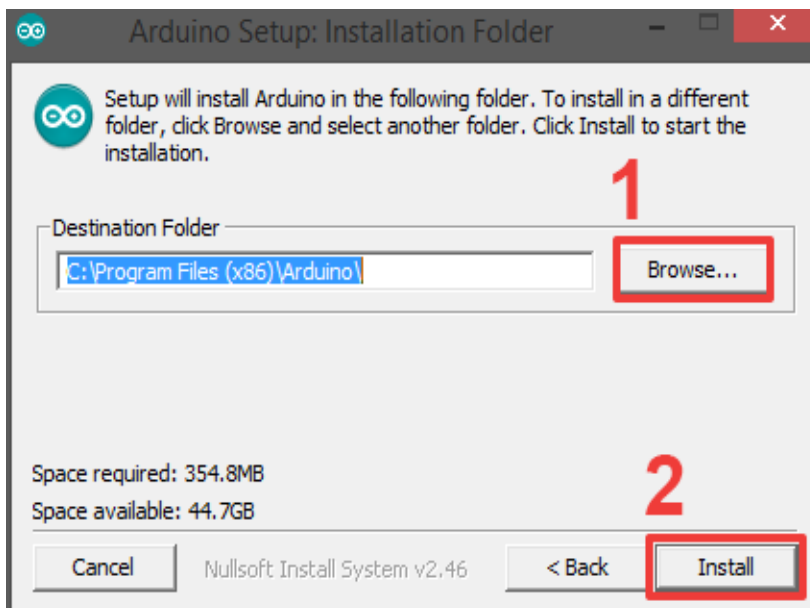
Joonis 6. Arduino IDE paigaldamine. Lepingutingimustega nõustumine.

2. Valida milliseid komponente paigaldatakse. Soovitav on paigaldada kõik komponendid, sest siis luuakse programmi otseteed Start menüüsse ja töölauale. Lisaks seotakse faililaiendiga “ino” ehk Arduino programmi failid Arduino arenduskeskkonnaga, mis võimaldab programmi avada otse Arduino arenduskeskkonnas. Pärast paigaldatavate komponentide valikut vajutada “Next >” nuppu (Joonis 7).



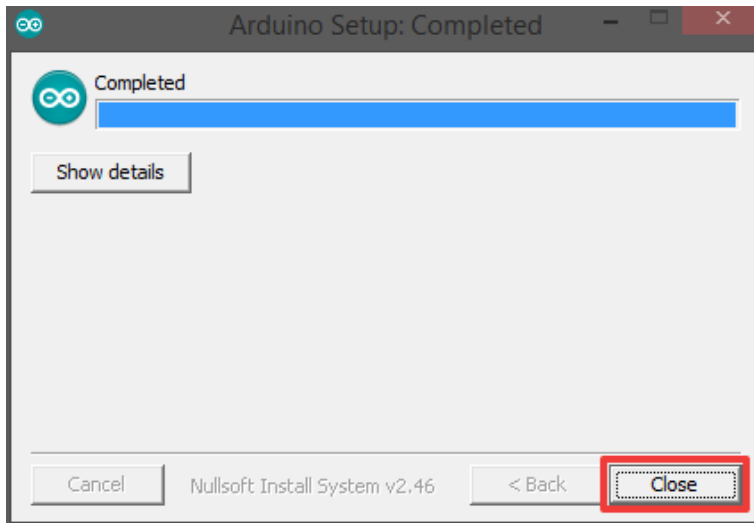
Joonis 7. Arduino IDE paigaldamine. Komponentide valimine.

3. Valida kaust, kuhu programm paigaldatakse. Vaikeväärtuse muutmiseks vajutada “Browse” nupule ja valida sobiv kaust. Paigaldusprotsessi alustamiseks vajutada nuppu “Install” (Joonis 8).



Joonis 8. Arduino IDE paigaldamine. Asukoha valimine ja paigaldusprotsessi alustamine.

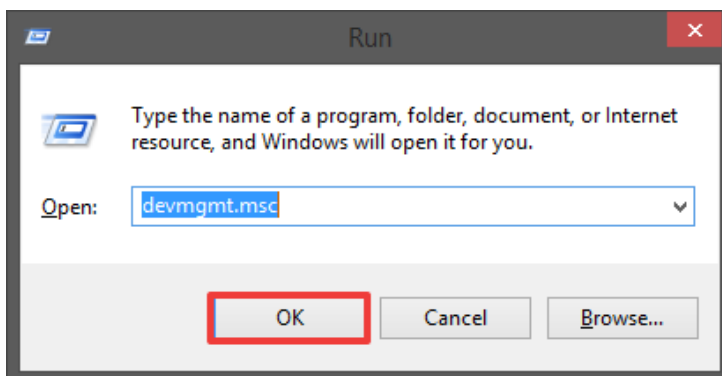
4. Kui tarkvara paigaldamine on lõppenud, vajutada nuppu “Close” (Joonis 9).



Joonis 9. Arduino IDE paigaldamine on edukalt lõppenud.

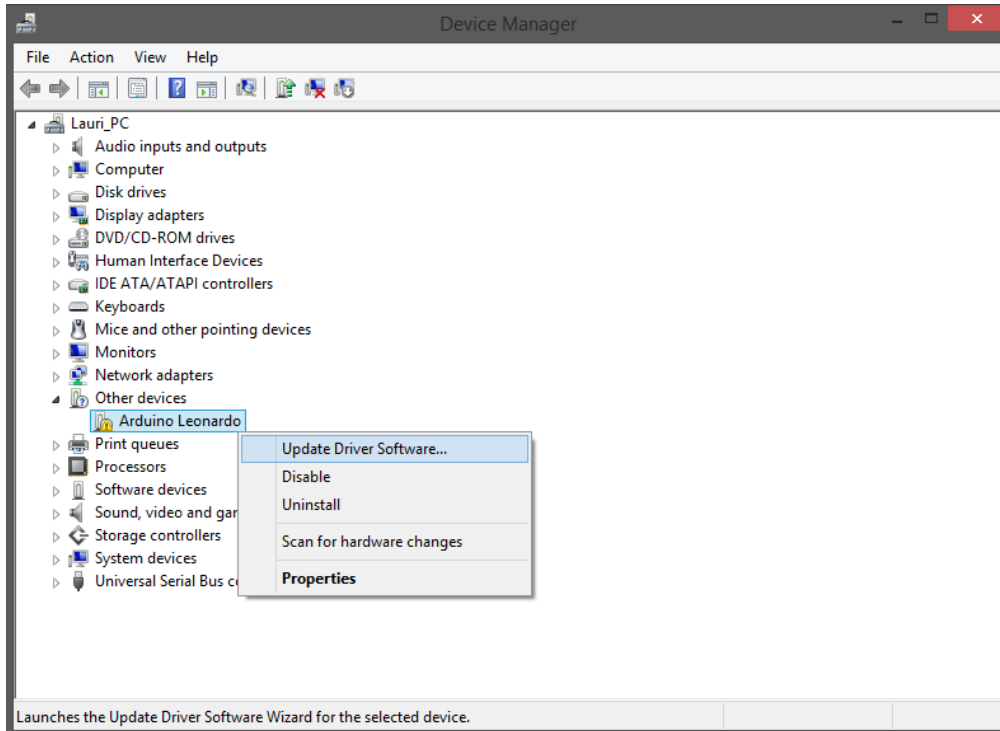
Lisaks Arduino arenduskeskkonnale on vaja paigaldada ka draiverid. Draiverite paigaldamiseks tuleb toimida järgmiselt:

1. Ühenda Arduino arendusplaat USB kaabli abil arvutiga.
2. Windows proovib automaatselt draiverid paigaldada. Kui automaatne draiverite paigaldus õnnestub, jätta järgmised kaheksa punkti vahele ja minna järgmise lõigu juurde, vastasel juhul minna käesoleva juhendiga edasi.
3. Vajutada klahvikombinatsiooni Win+R ja avanenud aknas olevale reale trükkida “devmgmt.msc” ning seejärel vajutada “OK” nupule (Joonis 10). Avaneb süsteemi seadmehaldur.



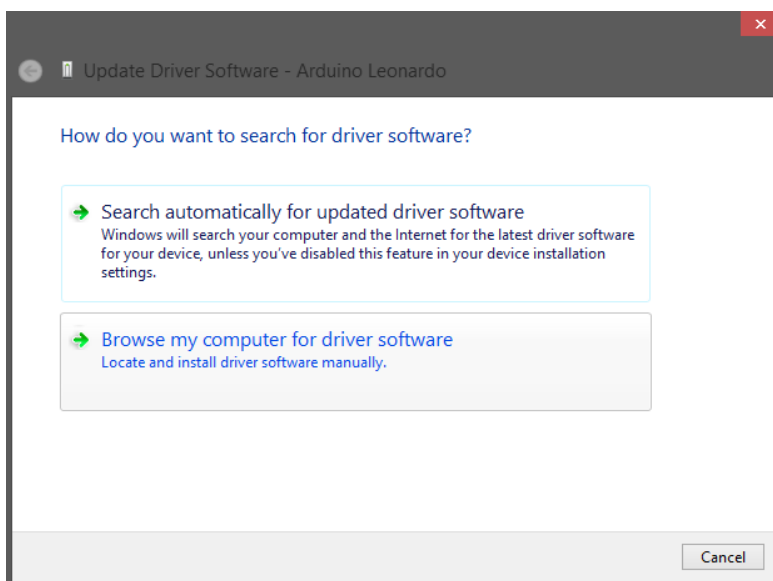
Joonis 10. Windows seadmehalduri avamine.

4. Vaadate “Ports (COM & LPT)” alla. Seal peaks olema avatud port nimega “Arduino Leonardo (COMx)”, kus x on mingi number. Kui “Ports (COM & LPT)” sektsiooni ei eksisteeri, siis vaadate “Other devices” või “Unknown devices” alla.
5. Teha paremklõps “Arduino Leonardo (COMx) peal ja valida “Update Driver Software...” (Joonis 11).



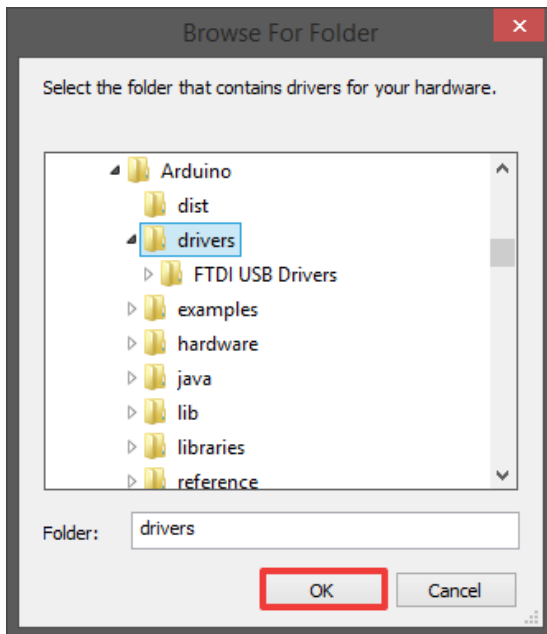
Joonis 11. Arduino Leonardo draiverite paigalduse alustamine.

6. Avanened aknas valida “Browse my computer for driver software” (Joonis 12).



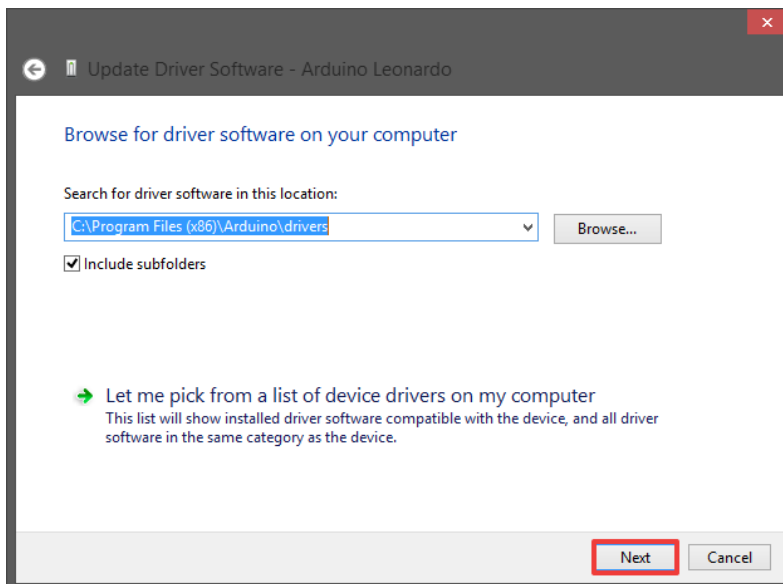
Joonis 12. Arduino Leonardo draiverite paigaldamine.

7. Vajutada nupul “Browse” ja navigeerida kausta kuhu Arduino tarkvara eelnevalt paigaldati. Sellest kaustast valida “Drivers” kaust (Joonis 13).



Joonis 13. Arduino Leonardo draiverite paigaldamine.

8. Vajutada “Ok” nupul (Joonis 13).
9. Vajutada “Next” nupul (Joonis 14).

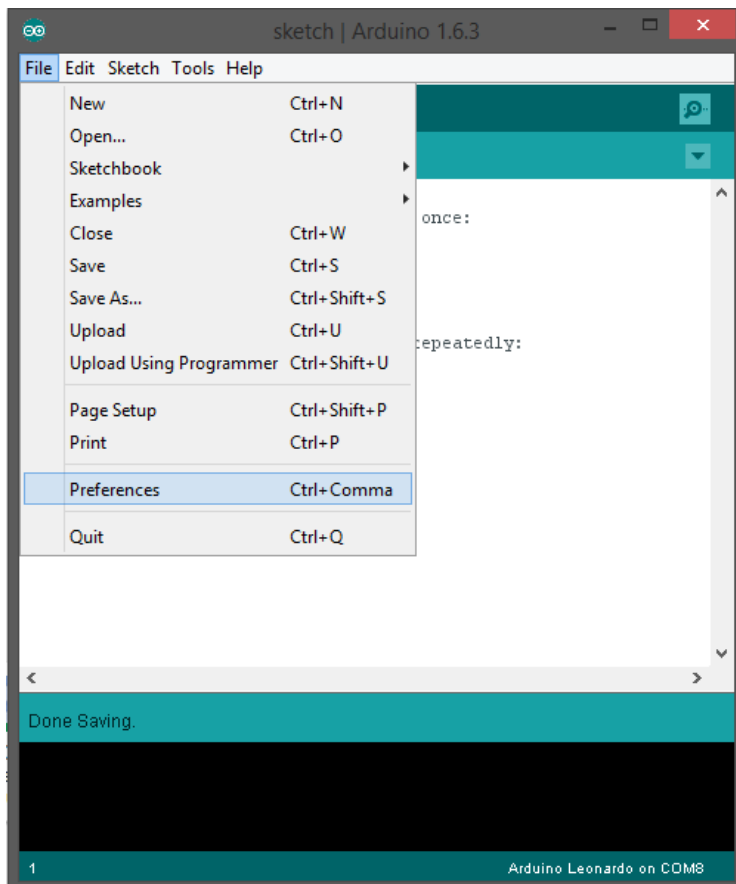


Joonis 14. Arduino Leonardo draiverite paigaldamine.

10. Draiver on paigaldatud. Vajutada “Close” nupul.

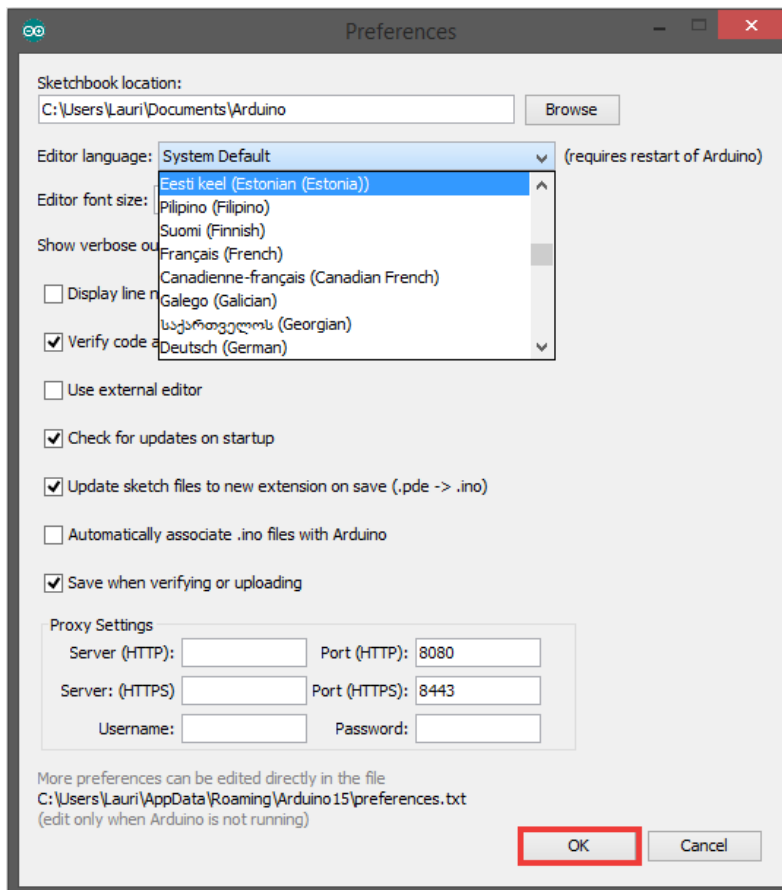
Arduino arenduskeskkonna ja USB draiveri paigaldamise juhised on sellega lõppenud. Järgmisena kirjeldatakse Arduino arenduskeskkonna kasutajaliidese keele muutmist ning lihtsa valgusdiodi vilgutamise programmi laadimist Arduino arendusplaadi mällu.

Käivitada Arduino arenduskeskkond avades töölaual olev Arduino programmi otsetee. Esmalt muuta arenduskeskkonna kasutajaliides eestikeelseks. Selleks avada eelistuste aken valides “File” menüüst “Preferences” (Joonis 15).



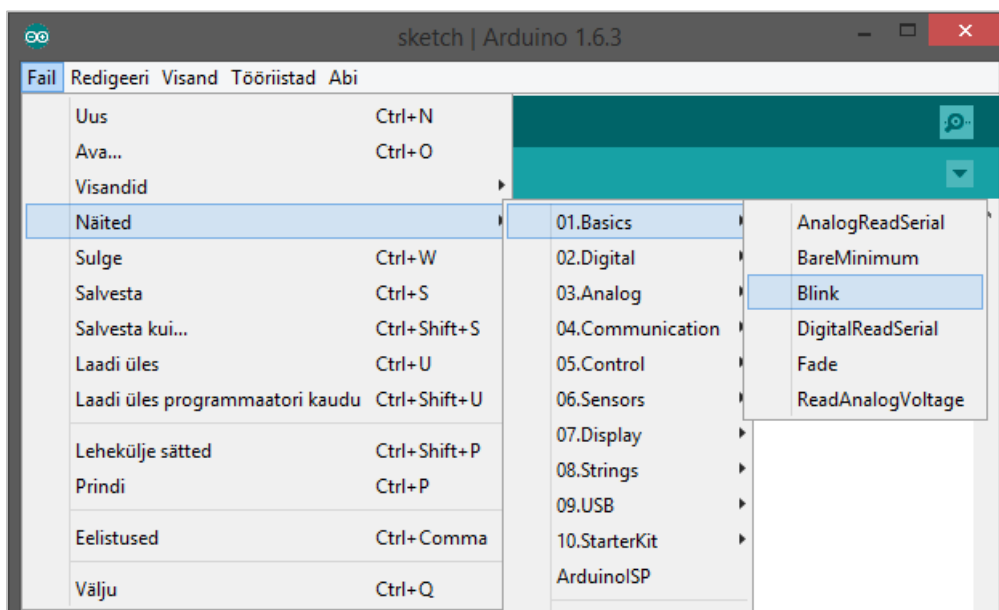
Joonis 15. Arduino arenduskeskkonnas eelistuste akna avamine.

Avanened aknas valida “Editor language” väärtuseks “Eesti keel” ning vajutada “OK” nupul (Joonis 16). Seejärel taaskäivitada Arduino arenduskeskkond.



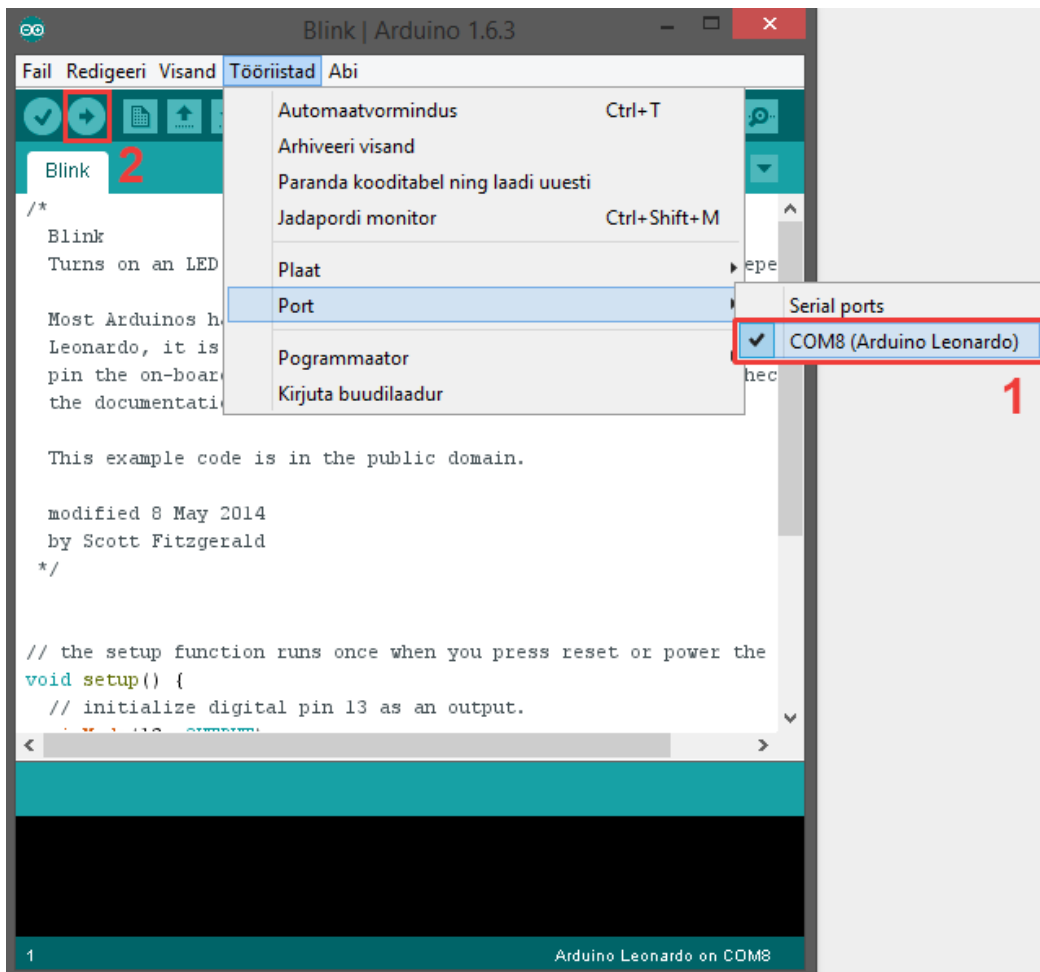
Joonis 16. Arduino arenduskeskkonna kasutajaliidese keele muutmine.

Valida “Fail” menüüst “Näited” -> “Basics” -> “Blink” nagu on näidatud joonisel 17. Avaneb uus aken, kus on valgusdiodi vilgutamise programm.



Joonis 17. Blink programmi valimine Arduino arenduskeskkonnas.

Kuna programmi kasutatakse esimest korda, tuleb seadistada tööks õige arendusplaat ja port. Selleks valida “Tööriistad” menüüst “Plaat” -> “Arduino Leonardo” ning “Port” -> “COMx (Joonis 18), kus x tähistab sama numbrit, mida näitas eelnevalt ka seadmehaldur. Kui pole kindel millist COM porti valida, siis võib Arduino arendusplaadi arvuti küljest korraks lahti ühendada ja vaadata milline port menüüst ära kadus. Arendusplaadi uuesti ühendamisel tulebki see port valida. Üldjuhul on selleks pordiks nimekirjas olev suurima numbriga port.



Joonis 18. Arduino arenduskeskkonnas õige pordi valimine.

Seejärel vajutada üleval vasakus nurgas asuvale noolele (“Laadi üles”) (Joonis 18), et valitud programm Arduino Leonardo mällu laadida. Peaks ilmuma laadimisriba ja tekst “Visandi kompileerimine” ja seejärel “Üleslaadimine...” ning Arduino Leonardo peal peaksid hakkama RX ja TX tulukesed vilkuma. Kui antud programm sai edukalt Arduino Leonardo mällu laetud, hakkab “L”-tähe juures olev valgusdiod (Joonis 19) iga sekundi tagant vilkuma. Seega valgusdiodi vilgutamise programm töötab ning kõik on õigesti seadistatud.



Joonis 19. Arduino Leonardo sissehitatud valgusdiodi vilkumine.

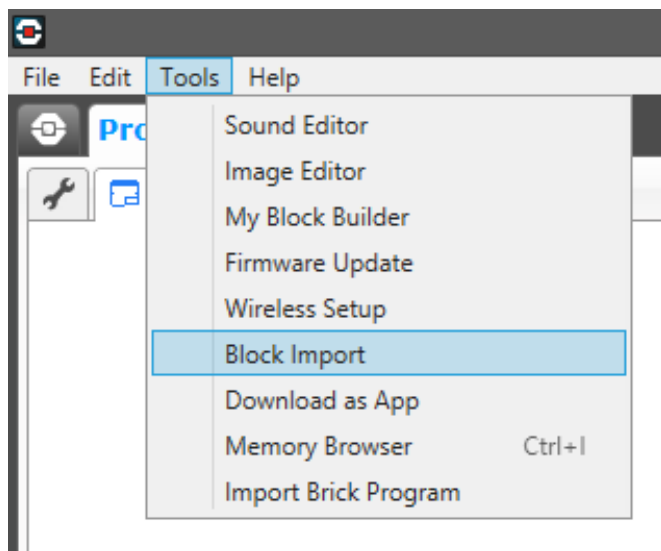
Selles punktis anti juhised Arduino arenduskeskkonna ja draiverite paigalduseks ning lihtsa valgusdiodi vilgutamise programmi käivitamiseks Arduino arendusplaadil. Järgmises punktis antakse ülevaade EV3 juhtplokkide vajaminevast tarkvarast.

2.3.2 EV3 tarkvara Arduino arendusplaadiga ühendamiseks

LEGO Mindstorms EV3 programmeerimiseks läheb vaja EV3 tarkvara, mille saab alla laadida LEGO Mindstorms EV3 kodulehe “Downloads” sektsioonist (<http://www.lego.com/en-us/mindstorms/downloads/download-software>).

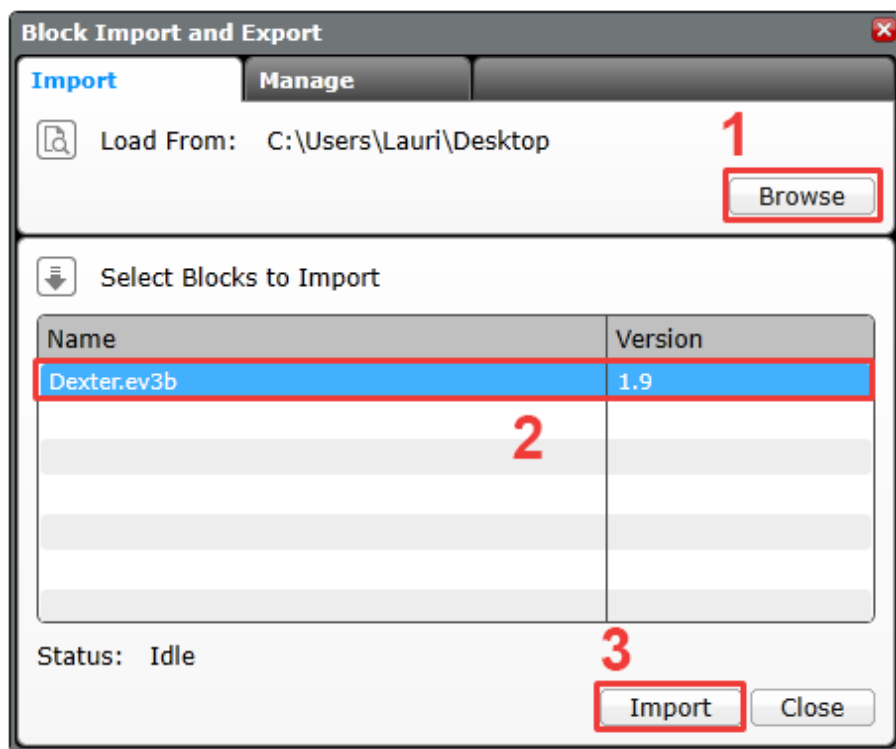
Lisaks sellele läheb vaja ka firma Dexter Industries poolt loodud plokkide EV3 tarkvara jaoks. Need plokiid sisaldavad antud ühenduse jaoks vajalikku I²C kommunikatsiooni plokkide. Nimetatud plokiid saab alla laadida Dexter Industries kodulehelt “EV3” sektsioonist või github repositooriumist (https://github.com/DexterInd/EV3_Dexter_Industries_Sensors/raw/master/Dexter.ev3b).

Alustuseks tuleb Dexter.ev3b importida LEGO Mindstorms EV3 tarkvara keskkonda. Selleks valida menüürilt “Tools” -> “Block Import” (Joonis 20).



Joonis 20. EV3 keskkonda uue ploki importimise alustamine.

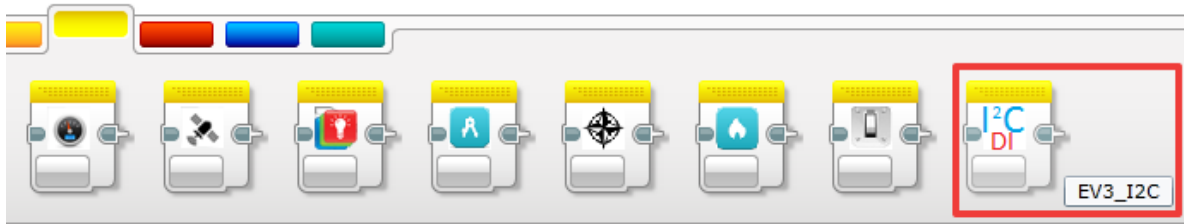
Avanenud aknas vajutada “Browse” nupule, navigeerida kausta, kus eelnevalt alla laetud fail “Dexter.ev3b” asub ning valida see fail. Imporditavate plokkide nimekirjas valida “Dexter.ev3b” ning vajutada nupule “Import” (Joonis 21).



Joonis 21. EV3 keskkonda uue ploki importimine.

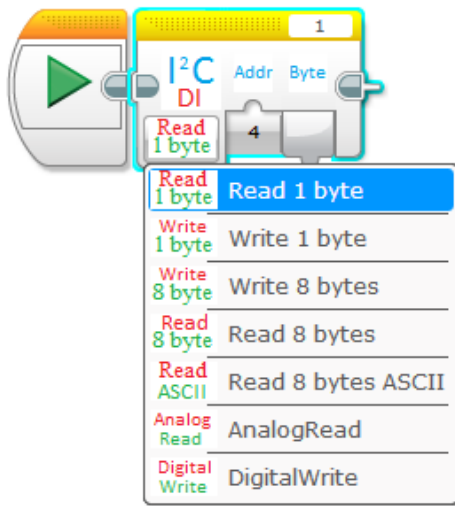
Seejärel tuleb LEGO Mindstorms EV3 tarkvara taaskäivitada, et imporditud plokke kasutada.

Eelnevalt imporditud plokid asuvad EV3 keskkonnas kollase “Sensor” vahekaardi all. Nende hulgast läheb vaja plokki nimega “EV3_I2C“ (Joonis 22).

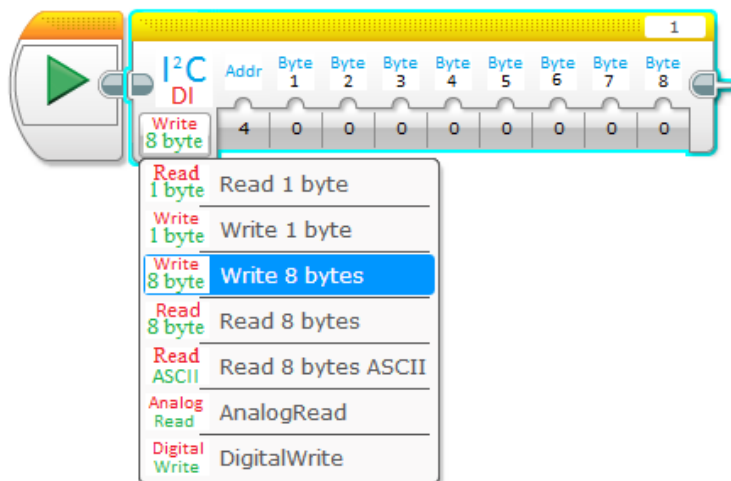


Joonis 22. Dexter Industries I2C kommunikatsiooniplokk LEGO Mindstorms EV3 tarkvara jaoks.

Eelnimetatud plokk võimaldab lugeda ja kirjutada 1 baidi kaupa (Joonis 23) või 8 baidi kaupa (Joonis 24).

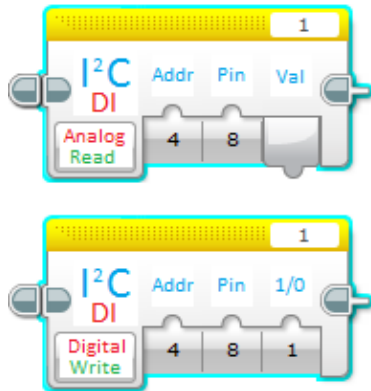


Joonis 23. Dexter Industries I²C kommunikatsiooniploki funktsioonid. Valitud on 1 baidi kaupa kirjutamine.



Joonis 24. Dexter Industries I²C kommunikatsiooniploki 8 baidi kaupa kirjutamise funktsioon.

Samuti saab kasutada I²C kommunikatsiooniploki funktsioone DigitalWrite ja AnalogRead (Joonis 25), mis vastavad Arduino funktsioonidele DigitalWrite(viik, väärtus) ja AnalogRead(viik).



Joonis 25. Dexter Industries I²C kommunikatsiooniploki DigitalWrite ja AnalogRead funktsioonid.

Iga funktsiooni kasutades tuleb märkida ka “Addr” väljale aadress. See on alamseadme aadress ning peab ühtima Arduino programmis defineeritud aadressiga.

Käesolevas punktis tutvustati EV3 juhtploki jaoks vajaminevat tarkvara. Järgmises punktis antakse ülevaade Arduino arendusplaadi ja EV3 juhtploki omavahelise ühendamise jaoks vajaminevast riistvarast.

2.4 Riistvara Arduino ja LEGO Mindstorms EV3 ühendamiseks

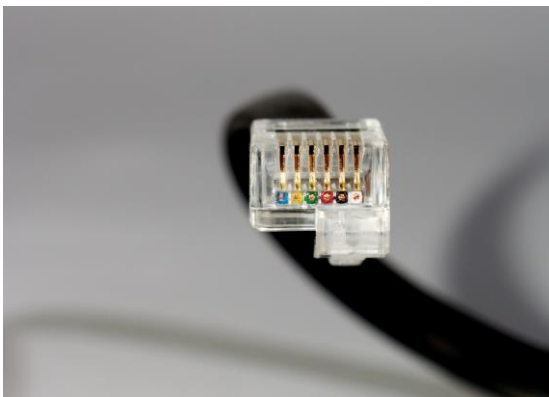
Arduino arendusplaadi ühendamiseks LEGO Mindstorms EV3 juhtploki kasutatakse EV3 juhtploki sensori porti. Järgnevalt on toodud juhised vajaliku ühenduse tegemiseks. Järgnev juhend tugineb firma Dexter Industries loodud õpetusel EV3 ja Arduino ühendamiseks [17].

Vaja läheb:

1. EV3 juhtplokk,
2. Arduino arendusplaat (antud juhendis Arduino Leonardo),
3. LEGO Mindstorms ühenduskaabel,
4. makettlaud,
5. makettlaua ühendusjuhtmed.

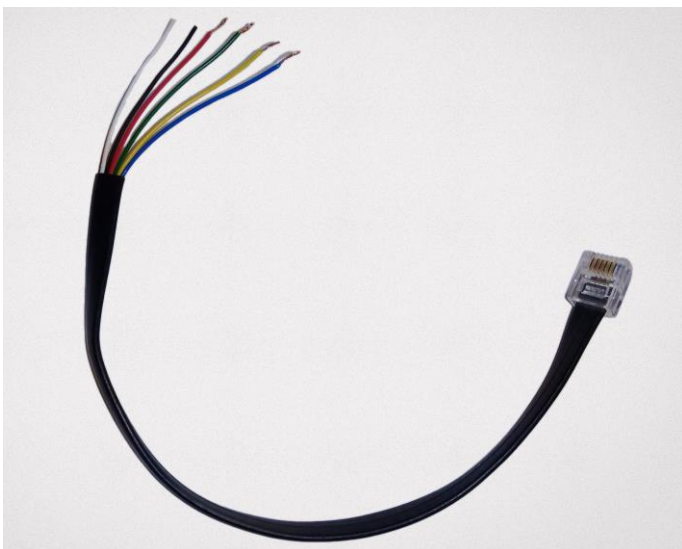
Arduino arendusplaat ühendatakse EV3 juhtploki külge LEGO Mindstorms ühenduskaabli abil. Ühenduskaablis (Joonis 26) on järgmised juhtmed:

1. valge - analoog (ANA)
2. must - maandus (GND)
3. punane - maandus (GND)
4. roheline - toide (V_{CC})
5. kollane - I²C siini taktsignaali (SCL)
6. sinine - I²C siini andmesignaali (SDA)



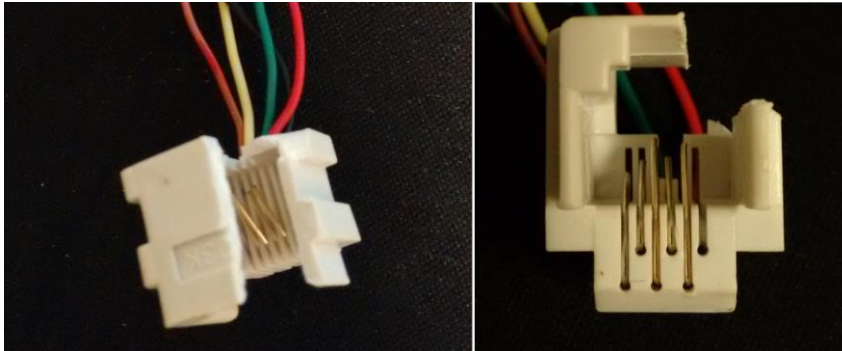
Joonis 26. LEGO Mindstorms ühenduskaabel [18].

Juhtmete ühendamiseks on mitu võimalust. Kõige lihtsam on kasutada Dexter Industries poolt loodud makettlauda adapterit (<http://www.dexterindustries.com/site/?product=breadboard-adapter-lego-mindstorms#prettyPhoto>). Kui adapterit käepärast pole, võib lõigata ühenduskaabli ühe otsa maha ja paljastada juhtmete otsad (Joonis 27).



Joonis 27. LEGO Mindstorms ühenduskaabel, mille ühes otsas on juhtmed paljastatud.

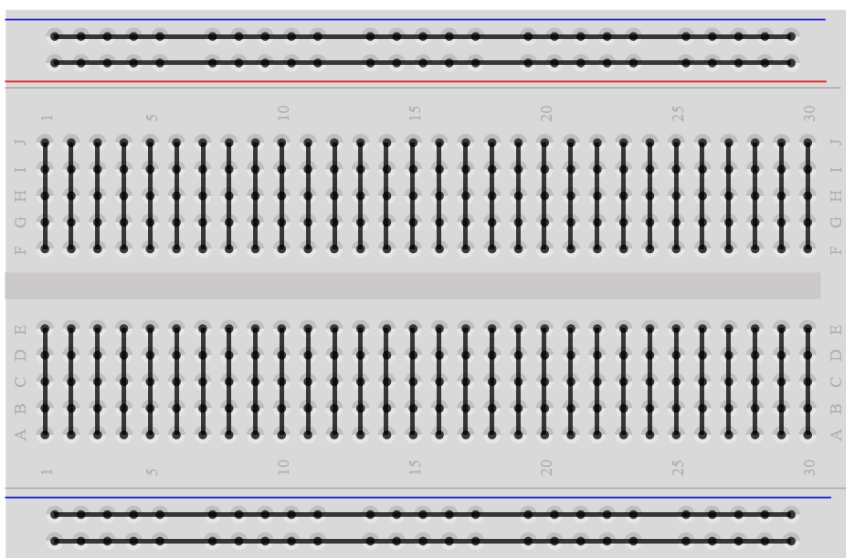
Kui aga kaablit mingil põhjusel lõigata pole võimalik, saab vastava adapteri ise meisterdada. Selleks tuleb standardisel RJ-12 pesal üks nurk maha lõigata nagu näidatud joonisel 28 ning seejärel saab LEGO Mindstorms ühenduskaabli RJ-12 pesaga ühendada.



Joonis 28. LEGO Mindstorms kaabliga ühendamiseks modifitseeritud RJ-12 pesa.

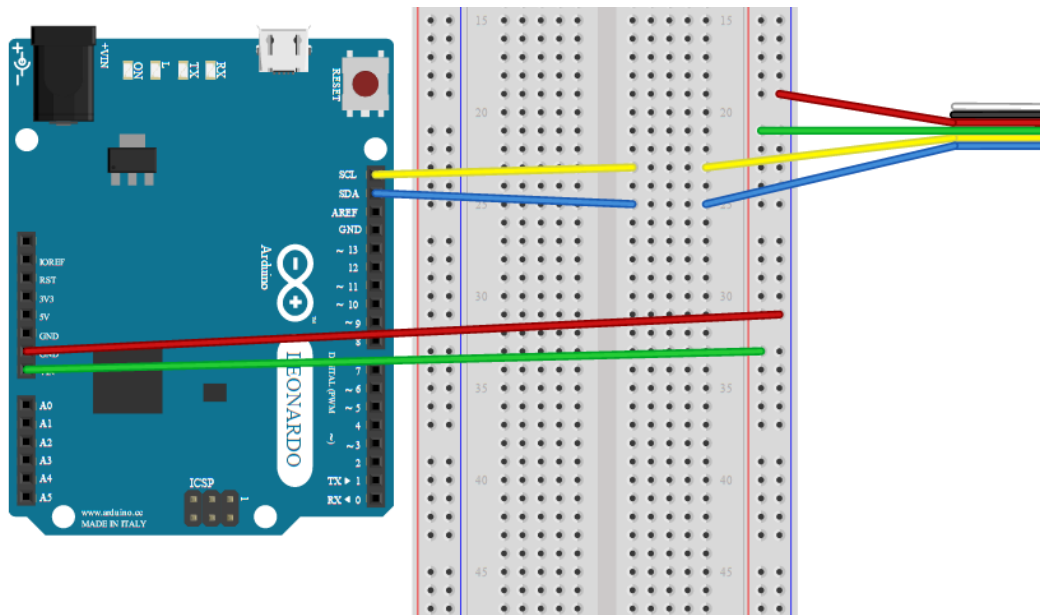
Arduino arendusplaadiga on vaja ühendada sinine SDA juhe, kollane SCL juhe ning tööpinge ühtlustamiseks ka punane GND juhe. Kui tekib vajadus Arduino arendusplaadi vooluallikana kasutada sellesama kaabli abil EV3 juhtplokki, siis tuleb ühendada ka roheline V_{CC} juhe. Arduino toiteallikatest on pikemalt kirjutatud punktis 2.5. SDA juhe tuleb ühendada Arduino SDA pessa, SCL juhe Arduino SCL pessa, V_{CC} juhe Arduino V_{in} pessa ning GND juhe Arduino GND pessa. Valget ja musta juhet kuhugi ühendama ei pea.

Selguse mõttes on hea ühendada juhtmed maketlauru kaudu. Maketlauru omavahel ühenduses olevad pesad on kujutatud joonisel 29.



Joonis 29. Maketlauru omavahel ühenduses olevad pesad.

EV3 kaablist tulevate juhtmete ühendamise skeem on kujutatud joonisel 30.



Joonis 30. EV3 juhtmete ühendamine Arduino Leonardo külge.

Siinkohal peaks veel ära märkima, et kui I²C ühendus mingil põhjusel ei tööta, siis peab ühendama ka 47k Ω takistid SDA juhtme ja V_{CC} juhtme vahele ning SCL juhtme ja V_{CC} juhtme vahele. Lisaks saab näiteks Arduino Leonardol “SDA” ja “SCL” viikudena kasutada ka vastavalt “2” ja “3” viiku, seega ei tohi nende viikude külge midagi muud ühendada. Erinevate Arduino arnedusplaatide I²C viikude asukohad on leitavad Arduino kodulehelt toote tutvustuse alt (<http://www.arduino.cc/en/Main/Products>).

Peale vajalike juhtmete ühendamist Arduino arendusplaadi külge, tuleb kaabli teine ots ühendada EV3 juhtploki sensori porti ehk porti, mis on nummerdatud 1-4.

Kirjeldati kuidas riistvara poole pealt Arduino ühendada LEGO Mindstorms EV3 juhtploki. Järgmises punktis kirjeldatakse erinevaid võimalusi Arduino arendusplaadi toitmiseks

2.5 Toiteallikad Arduino arendusplaadi jaoks

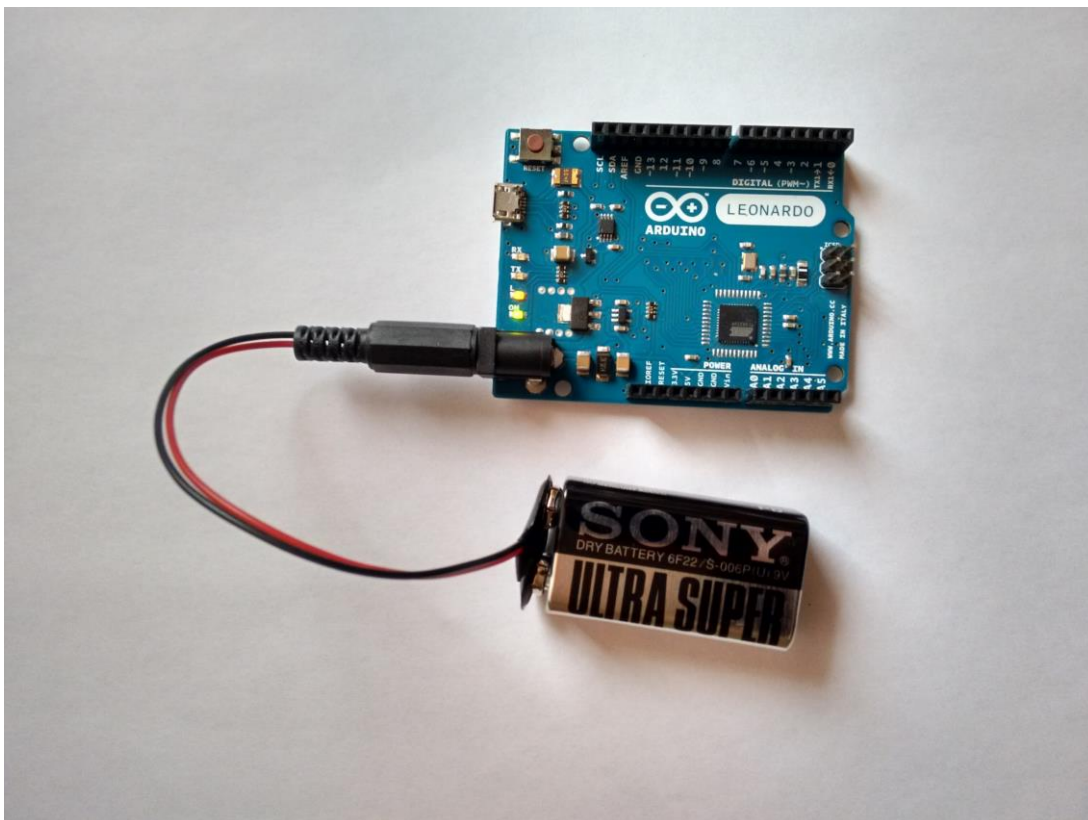
Arduino arendusplaat saab toite USB liidese kaudu, eraldi toitesisendist või V_{in} viigu kaudu. Järgnevalt on kirjeldatud kõiki kolme võimalust Arduino arendusplaadi toitmiseks.

LEGO Mindstorms EV3 juhtploki sensori porti ühendatud kaabli roheline V_{CC} juhtme väljundpinge on 5V, mis on Arduino arendusplaadile sobiv. Samuti on voolutugevus piisav, et

Arduino arendusplaat ära toita. Kuna EV3 juhtploki tulevad sinine, kollane ja punane juhe on niikuinii vaja ühendada Arduino arendusplaadi külge, siis pole eriti keeruline ka rohelist V_{CC} juhet arendusplaadi V_{in} viiguga ühendada.

Veel üks võimalus on Arduino arendusplaati toita USB liidese kaudu. Seda on võimalik teha mitmel viisil. Näiteks saab kasutada EV3 juhtploki toiteallikana. Nimetatud lahendus on väga mugav – USB juhtme üks ots tuleb ühendada arendusplaadi USB porti ning teine ots EV3 juhtploki A-tüüpi USB porti ehk mälukaardi pesa kõrval asuvasse USB porti. Kui aga oluline on EV3 juhtploki akut säästa, siis võib USB juhtme ühendada välise toiteallikaga nagu näiteks USB akupank.

Lisaks eelnevale on võimalik kasutada ka Arduino arendusplaadil olevat eraldi toitesisendit. Näiteks vastava juhtme abil saab toiteallikana kasutada 9-voldist patareid (Joonis 31). See lahendus on mõistlik jällegi siis, kui on vaja EV3 juhtploki akut säästa.



Joonis 31. Arduino Leonardo toide 9-voldisest patareist.

Lisatoiteallikaid on mõistlik kasutada ka siis, kui Arduinoga ühendatakse mootoreid või muid seadmeid, mis töötavad suurel võimsusel.

Antud punktis kirjutati Arduino arendusplaadi toitevõimalustest. Järgmises punktis antakse juhised EV3 juhtploki ja Arduino arendusplaadi vahelise ühenduse testimiseks.

2.6 Arduino ja EV3 vahelise ühenduse testimine

Käesolevas punktis kirjeldatakse kuidas Arduino ja EV3 vahelist ühendust testida ning missugused programmid selle jaoks kummalegi seadmele koostada. Lõpptulemusena saadetakse Arduino jadapordi monitori aknasse EV3 juhtploki vajutatava nupu ID ning EV3 juhtploki ekraanil kuvatakse Arduino jadapordi monitori aknasse sisestatud sümboli ASCII väärtus.

Järgnevate juhiste abil saab proovida, kas suhtlus LEGO Mindstorms EV3 ja Arduino arendusplaadi vahel toimib. LEGO Mindstorms EV3 juhtploki suhtluseks kasutatakse I²C siini. EV3 on antud I²C seadistuses ülemseade ja Arduino alamseade. Kõigepealt näidatakse kuidas antud testi jaoks Arduino programm koostada.

Alustuseks on vaja käivitada Arduino arenduskeskkond. Kuna Arduino ja EV3 vaheline suhtlus käib I²C siini abil, siis tuleb valida alamseadme aadress. See aadress peab ühtima EV3 programmis oleva aadressiga. Antud näites seame selleks aadressiks 4. I²C suhtluseks kasutab Arduino teeki *Wire*. Järgnevalt on toodud testimiseks vajalik Arduino programm.

```
#include <Wire.h>
#define ALAMSEADME_ADDRESS 4 //seame alamseadme aadressiks 4

void setup() {
  //kood, mis jookseb üks kord Arduino käivitamisel
  Serial.begin(9600); //alustame suhtlust arvutiga kiirusel 9600 boodi
  Wire.begin(ALAMSEADME_ADDRESS); //kasutades teeki Wire, ühendu I2C
  siiniga kui alamseade
  Wire.onReceive(andmedSisse); //määrame funktsiooni, mis kutsutakse, kui
  ülemseade saadab andmeid
  Wire.onRequest(andmedValja); //määrame funktsiooni, mis kutsutakse, kui
  ülemseade nõuab andmeid
}
int vaartus = 0;
int lipp = 0;
int jadaMon = 0;

void loop() {
  //kood, mis jookseb lõputus tsüklis
  if (lipp == 1) { //lipp = 1, siis kui ülemseade saatis andmeid
    if (vaartus != 0) { //kui vajutatakse EV3 juhtploki nuppu (kui nuppu ei
    vajutata, saadab EV3 arvu 0)
      Serial.print("Vajutatakse EV3 nuppu, mille ID on "); //kirjuta
```

```

Arduino jadapordi monitori aknasse
    Serial.println(vaartus);
}
lipp = 0; //seame lipu tagasi 0, sest saadetud andmed on töödeldud
}
while (Serial.available() > 0) { //kontrollime kas jadapordi monitor on
saatnud andmeid
    jadaMon = Serial.read(); //loeme saadetud andmed
    Serial.print("EV3 ekraanil peaks olema: ");
    Serial.println(jadaMon);
}
}
//funktsioon, mis käivitatakse, kui ülemseade saadab andmeid
void andmedSisse(int baite) {
    while (Wire.available() > 0) { //loeme kõik andmed I2C siinist
        vaartus = Wire.read();
        lipp = 1; //seame lipu väärtuseks 1, sest lugesime uusi andmeid
    }
}
// funktsioon, mis käivitatakse, kui ülemseade nõuab andmeid
void andmedValja() {
    Wire.write(jadaMon); //saadame jadapordi monitoris sisestatud väärtuse
}

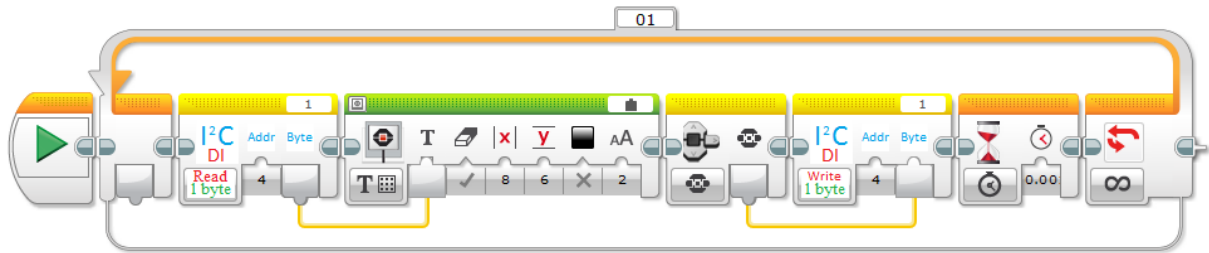
```

Antud näites funktsioon *setup()* käivitab arvutiga suhtluseks jadapordi boodikiirusega (ingl k. *baud rate*) 9600 boodi, määrab alamseadme aadressi ja funktsioonid, mida kutsutakse kui andmeid on vaja vastu võtta või saata. Kui saadetakse andmeid, siis kuvatakse iga saadetud bait jadapordi monitori aknasse ning kui nõutakse andmeid, siis saadetakse jadapordi monitoris viimasena sisestatud sümboli ASCII väärtus. Programmifail asub lisana failis test1.ino.

Ülaltoodud programm tuleb laadida Arduino arendusplaadi mällu. Selleks ühendada Arduino arendusplaat arvutiga, veenduda, et Arduino arenduskeskkonnas on valitud õige arendusplaadi mudel ja COM port ning vajutada “Laadi üles” nupule. Arduino arendusplaat jätta arvutiga ühendusse.

Sellega on Arduino arendusplaat testimiseks programmeeritud. Edasi näidatakse, milline näeb välja EV3 programm antud ühenduse testimiseks.

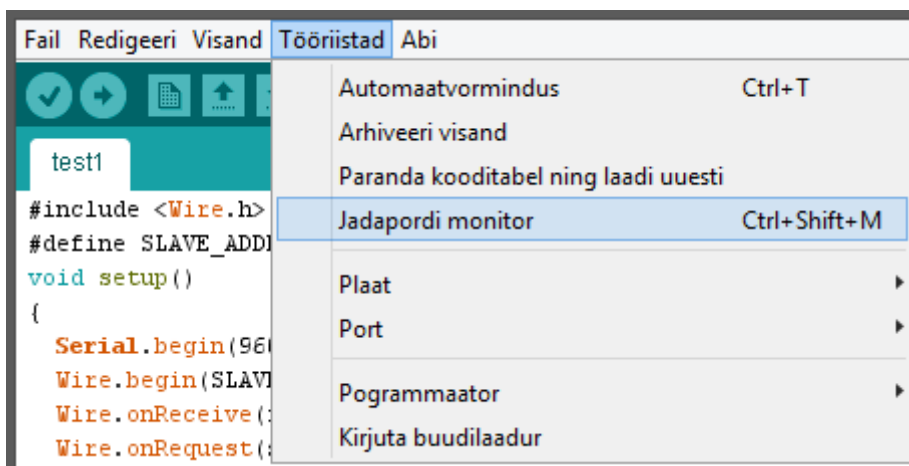
EV3 testprogrammi koostamisel kasutatakse tsüklit. Testprogramm on kujutatud joonisel 32 ning asub lisana failis test1.ev3s.



Joonis 32. EV3 testprogramm Arduino arendusplaadiga ühenduse testimiseks.

Tsükli keha koosneb viiest ploki. EV3_I2C ploki tuleb valida üleval paremas nurgas porti number, mis vastab EV3 juhtploki portile, kuhu on ühendatud Arduino arendusplaat. Esimese ploki funktsiooniks on ühe baidi lugemine. Loetud bait saadetakse järgmisele ploki, mis saadud baidi ekraanil kuvab. Järgmine plokk loeb millist nuppu EV3 juhtploki vajutatakse ja saadab selle nupu ID omakorda järgmisele ploki, mis omakorda saadab vajutatud nupu ID Arduino arendusplaadile. Soovitav on kuhugi tsükli sisse panna ka viivitus. Vastasel juhul koormatakse I²C liides üle ja ja ühendus ei toimi korrektselt. Viivituse pikkuseks sobib näiteks 0,001 sekundit.

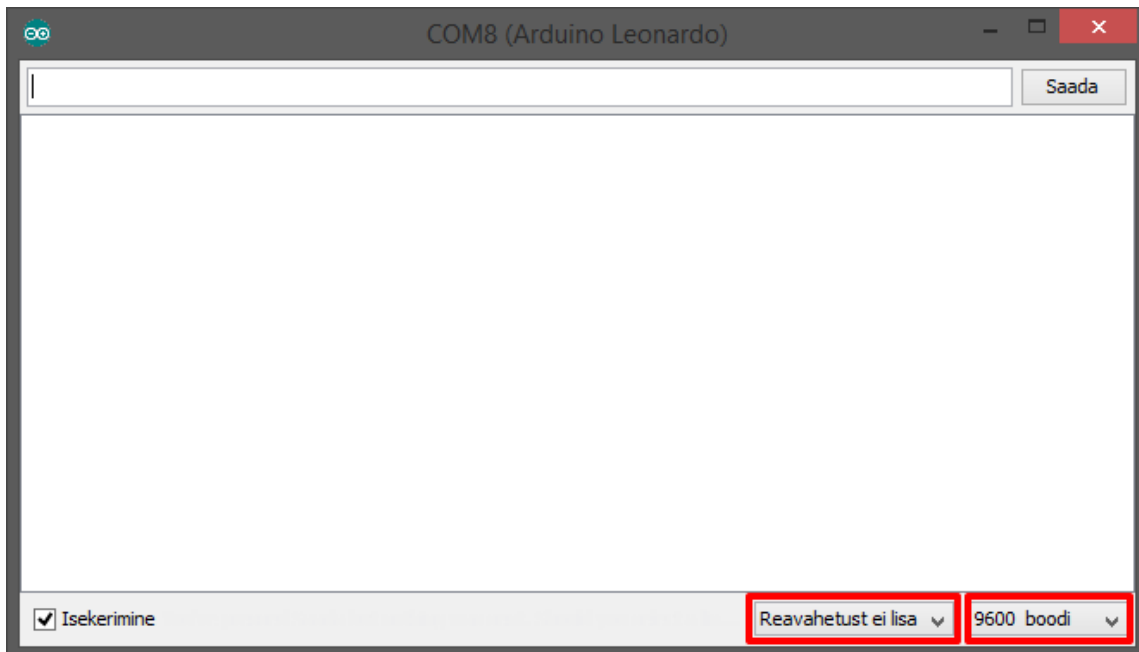
Arduino arendusplaadi ja EV3 vahelise suhtluse testimiseks käivitada EV3 juhtploki joonisel 32 toodud programm. Eelnevalt laeti Arduino programm juba arendusplaadi mällu ja Arduino jäi arvutiga ühendusse. Järgmiseks avada Arduino arenduskeskkonnas jadapordi monitor (Joonis 33).



Joonis 33. Arduino arenduskeskkonnas jadapordi monitori avamine.

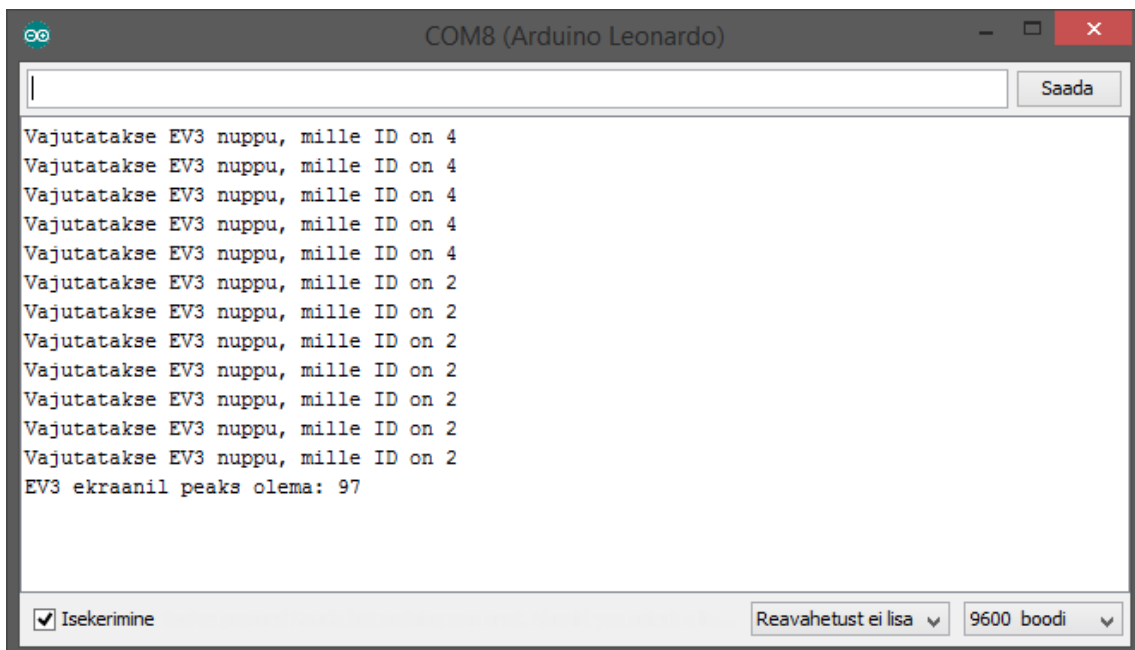
Avaneb jadapordi monitori aken. Enne jätkamist teha kindlaks, et akna alumises paremas nurgas oleks valitud “Reavahetust ei lisa” ja “9600 boodi” (Joonis 34), vastasel juhul saadetakse

EV3 juhtploki valeid andmed või ühendus jadapordi monitori ja Arduino arendusplaadi vahel ei toimi üldse või toimib vigaselt.



Joonis 34. Arduino arenduskeskkonna jadapordi monitor.

Kui nüüd EV3 juhtploki mõnda nuppu vajutada, peaks jadapordi monitor väljastama vajutatud nupu ID. Kui aga Arduino jadapordi monitori üleval asuvale reale trükkida mingi sümbol ja vajutada “Saada”, peaks EV3 juhtploki ekraanile ilmuma selle sümboli ASCII väärtus (Joonis 35).



Joonis 35. Arduino IDE jadapordi monitor EV3 ja Arduino vahelise ühenduse testimisel.

Kui jadapordi monitoris ja EV3 juhtploki ekraanil toimuv vastab eelnevalt kirjutatule, siis EV3 juhtploki ja Arduino arendusplaadi vaheline ühendus toimib.

Käesolevas peatükis anti soovitusi Arduino arendusplaadi valimisel ning toodi juhised Arduino ja LEGO Mindstorms EV3 juhtploki ühendamiseks ja ühenduse töö testimiseks. Samuti kirjutati Arduino arendusplaadi toitevõimalustest. Järgmises peatükis tuuakse viis erineva raskusastmega ülesannet, mida saab Arduino ja EV3 abil lahendada.

3. Ülesanded

Käesolevas peatükis antakse viis suureneva raskusastmega ülesannet LEGO Mindstorms EV3 ja Arduino kooskasutamiseks. Igal ülesandel on toodud tase, eesmärk, ülesande täitmiseks vajalike vahendite loetelu, ülesande püstitus, lahenduse idee, ülesande lahendus, loetelu tekkida võivatest probleemidest ning alternatiivid antud ülesandele. Lahenduse idees selgitatakse uute kasutatavate vahendite tööpõhimõtet. Igas ülesandes kasutatakse mõnda uut Arduino funktsiooni, ka seda tutvustatakse lahenduse idees. Samuti antakse igas ülesandes uusi teadmisi ning viimases ülesandes võetakse kokku kõik eelnevates ülesannetes õpitu, et ehitada puldist juhitud auto.

Siinkohal tuleb pöörata tähelepanu EV3 programmis kasutatavale EV3_I2C plokile. Nimelt, kui seda kasutatakse lõputu tsükli sees, peab sellesama tsükli sees olema ka viivituse plokk viivitusega näiteks 0,001s. Viivitus on piisavalt väike (tsükkel töötab 1000 korda sekundis), kuid ei koorma I²C liidest üle. Arvestada ka sellega, et Arduino arendusplaadil võib olla mitu paari SDA ja SCL viike. EV3 juhtploki ühenduses olles ei saa neid viike muuks otstarbeks kasutada.

3.1 Ülesanne 1 - valgusdiodi vilgutamine

Tase: lihtne

Eesmärk: panna valgusdiodid iga sekundi tagant vilkuma. Tutvuda DigitalWrite funktsiooniga ja Arduino programmeerimisega.

Ülesande täitmiseks vajalikud vahendid:

- Punktis 2.4 toodud juhiste järgi ühendatud LEGO Mindstorms EV3 juhtplokk ja Arduino arendusplaat
- Valgusdiodid
- 150Ω takisti
- Makettlaud
- Makettlaua juhtmed
- Arvuti LEGO Mindstorms EV3 tarkvaraga, millesse imporditud firma Dexter Industries I²C plokk, ja Arduino arenduskeskkonnaga

- Ülesannete lahendamiseks loodud Arduino programmi mall (Lisana failis mall.ino)

Ülesande püstitus:

Panna valgusdiodid iga sekundi aja tagant vilkuma.

Lahenduse idee:

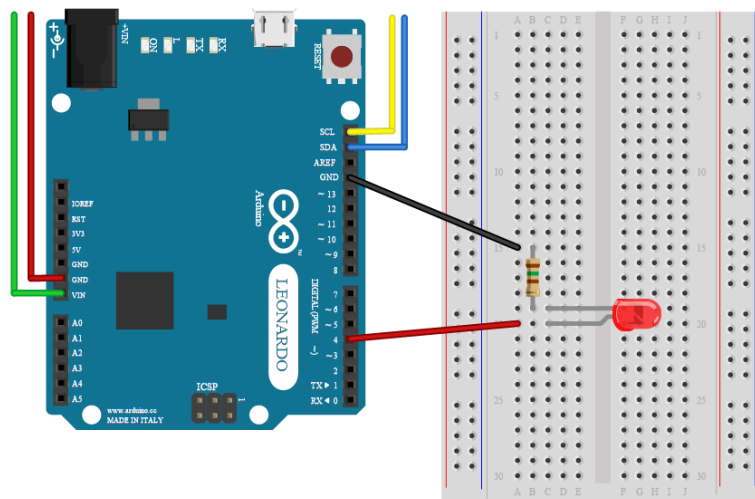
Ühendada valgusdiodid makettlauda külge. Valgusdiodil on kaks kontakti - anood ja katood. Kui rakendada anoodile positiivset pinget ja katoodile negatiivset pinget, siis valgusdiodid süttib, vastasel juhul mitte. Takisti on vajalik, sest arendusplaadist tulev 5V on valgusdiodi jaoks liiga palju ja kahjustab seda. Samuti on takistit vaja, et piirata arendusplaadi viigust tulevat voolutugevust. Voolutugevus, mis on suurem kui 40mA viigu kohta, kahjustab arendusplaati. Takisti võib ühendada anoodi külge või katoodi külge, sellest midagi ei muutu. Takisti ja valgusdiodi vabad kontaktid ühendada vastavalt Arduino arendusplaadi GND viigu ja digitaalse väljundiga viigu külge.

Arduino programmi mall on ette antud. Seda tuleb muuta, lisades õigesse kohta funktsioonid *pinMode(viik, väärtus)* ja *digitalWrite(viik, väärtus)*. Funktsioon *pinMode(viik, väärtus)* seab valitud viigu vastavalt sisendiks või väljundiks. Funktsioon *digitalWrite(viik, väärtus)* rakendab valitud väljundviigule pinge 5V (või 3,3V kui tegemist on 3,3V töötava Arduino arendusplaadiga), kui *väärtus* on *HIGH* või 1, ja pinge 0V (maandus), kui *väärtus* on *LOW* või 0. Lisaks tuleb muutujatesse lugeda EV3 poolt saadetakavad viigu number ja olek kasutades meetodit *Wire.read()*.

EV3 programmis saata tsükli sees Arduinole valitud viigu number ja olek, kasutades kahte *DigitalWrite* funktsiooniga I²C plokki. I²C plokkide vahele panna viivituse plokk.

Üks võimalik lahendus:

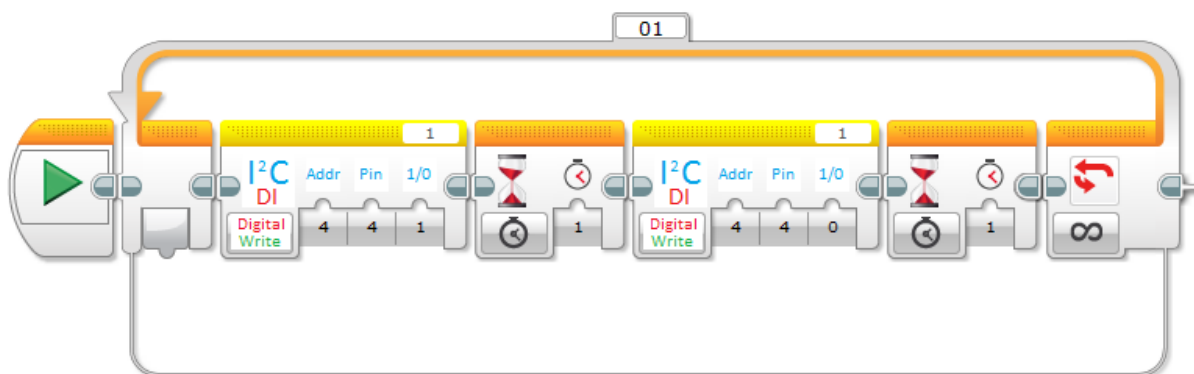
Ühendada valgusdiodi anood arendusplaadi digitaalse viiguga 4 ja katood läbi takisti arendusplaadi GND viiguga. Ühendamise skeem on kujutatud joonisel 36, kus sinine, kollane, punane ja roheline juhe suunduvad EV3 ühenduskaablistesse, mille teine ots on ühendatud EV3 juhtploki pordiga 1.



Joonis 36. Valgusdiodi ühendamise Arduino arendusplaadiga. Sinine, kollane, punane ja roheline juhe kuuluvad EV3 ühenduskaablisse, mis on ühendatud EV3 juhtploki.

Avada ülesannete jaoks loodud Arduino programmi mall, mis asub lisana failis mall.ino ning muuda seda. Lisada muutujad `int viik;` ja `int olek;`. Programmi tsükli `if` plokki lisada `read pinMode(viik, OUTPUT);` ja `digitalWrite(viik, olek);`. Funktsiooni `andmedSisse(int baite)` tsükli kehasse lisada kaks rida - `viik = Wire.read();` ja `olek = Wire.read();`. Need kaks rida loevad EV3 poolt saadetud viigu numbrit ja oleku vastavatesse muutujatesse. Näidislahendus on lisana failis y11.ino. Valminud programm laadida Arduino arenduspladi mällu.

Programm EV3 jaoks koosneb lõputust tsüklist. Tsüklistis tuleb lohistada kaks I²C plokki. Mõlema ploki ülemisest paremast nurgast määrata pordiks 2, "Addr" väärtuseks 4 ja "Pin" väärtuseks 4. Ühel plokil seada "1/0" väärtuseks 1 ja teisel 0. Pärast mõlemat I²C plokki lohistada tsüklistis ka viivituse plokk viivitusega 1 sekund. Programm on kujutatud joonisel 37 ja programmifail asub lisana failis y11.ev3s.



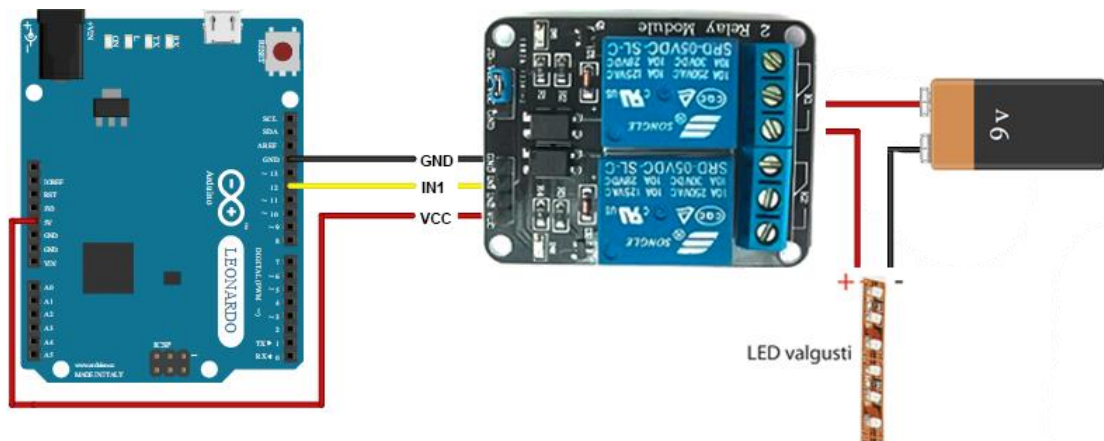
Joonis 37. Ülesande 1 EV3 programmi näide.

Tekkida võivad probleemid:

- “Addr” väärtus ei vasta Arduino programmis olevale “ALAMSEADME_ADDRESS” väärtusele.
Lahendus: seada „Addr“ väärtus ja “ALAMSEADME_ADDRESS” samaks.
- Määratud “Pin” väärtus ei vasta arendusplaadil valgusdiodi anoodiga ühenduses olevale viigule.
Lahendus: seada „Pin“ väärtus valstavalt arendusplaadil valgusdiodi anoodiga ühenduses olevale viigule.
- Valitud port ei vasta EV3 juhtploki pordile.
Lahendus: valida EV3 programmis õige port.

Ülesande alternatiivid:

- Ühendada mitme viigu külge valgusdiodid ja muuta EV3 programmi nii, et valgusdiodid vilguksid erinevas tempos.
- Ühendada arendusplaadi külge relee, mille abil saab välise toiteallika kaudu juhtida terveid valgusdiodi ribasid ja muid suurt voolutugevust nõudvaid seadmeid. Relee ühendamise skeem on toodud joonisel 38.



Joonis 38. Relee abil LED valgusti juhtimine [19].

3.2 Ülesanne 2 - valgusdiodi heleduse muutmine EV3 mootori abil

Tase: keskmine

Eesmärk: muuta valgusdiodi heledust lugedes EV3 keskmise mootori pöördenurka. Tutvuda I²C ploki võimalustega, Arduino `analogWrite()` funktsiooniga ja Arduino `map()` funktsiooniga.

Ülesande täitmiseks vajalikud vahendid:

- Punktis 2.4 toodud juhiste järgi ühendatud LEGO Mindstorms EV3 juhtplokk ja Arduino arendusplaat
- EV3 keskmine mootor
- Valgusdiod
- 150Ω takisti
- Makettlaud
- Makettlaua juhtmed
- Arvuti LEGO Mindstorms EV3 tarkvaraga, millesse imporditud firma Dexter Industries I²C plokk, ja Arduino arenduskeskkonnaga.
- Ülesannete lahendamiseks loodud Arduino programmi mall (Lisana failis `mall.ino`)

Ülesande püstitus:

Muuta valgusdiodi heledust vastavalt EV3 mootori pöördenurgale

Lahenduse idee:

Valgusdiodi heleduse muutmiseks kasutada Arduino funktsiooni `analogWrite()`, mis töötab vaid arendusplaadi PWM viikudega. PWM ehk pulsilaiusmodulatsiooni tööpõhimõte on järgmine - voolu lülitatakse suurel sagedusel sisse-välja, millega tekitatakse vooluimpulsid. Impulsside laiuse muutmisega reguleeritakse väljundpinget. Funktsioon `analogWrite()` teisendab argumendiks saadud täisarvu lineaarselt vahemikus 0-255 väljundviigule rakendatavaks pingeks 0V-5V.

EV3 programm mõõdab, mitu kraadi on juhtplokkiga ühendatud mootorit pööratud ning saadab tulemuse Arduinole. Dexter Industries I²C plokk aga võimaldab saata korraga 7 bitti ehk arve 0-127. Suuremate arvude saatmiseks tuleb saata kaks väikest arvu. Sellepärast on vaja jagada

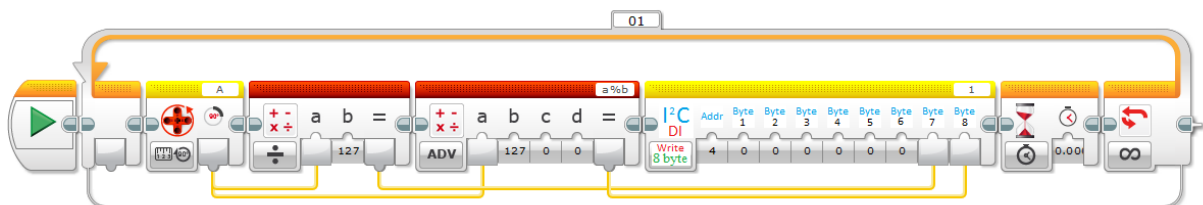
saadetav arv 127-ga ning seejärel leida saadetava arvu jääk jagamisel 127-ga. Saadud vastused saata üksteise järel Arduinole. Niiviisi on võimalik saata täisarve 0-16255.

Arduino programmis salvestada EV3 poolt saadetud kaks arvu. Esialgse arvu taastamiseks tuleb korrutada esimene arv 127-ga ning liita teine arv ehk jääk. Saadud pöördenurka kasutada näiteks siinusfunktsiooni argumentina, mille vastus hakkab sõltuvalt nurgast perioodiliselt korduma. Selleks, et *analogWrite()* argument oleks vahemikus 0-255, kasutada funktsiooni *map(sisend, x1, y1, x2, y2)*, mis teisendab muutuja *sisend* väärtused vahemikust *x1-y1* vahemikku *x2-y2*.

Üks võimalik lahendus:

Ühendada valgusdiodid Arduino arendusplaadiga nii nagu näidatud esimese ülesande lahenduses, ainult et viigu 4 asemal kasutada viiku 5, millel on PWM funktsionaalsus. EV3 juhtploki porti A ühendada mootor ning porti 1 ühendada Arduino arendusplaadiga ühenduses olev kaabel.

Koostada EV3 programm, milles on lõputu tsükkel. Tsüklisse lohistada “Motor rotation” plokk, kaks “Math” plokki, I²C plokk ja viivituse plokk. EV3 programm on toodud joonisel 39



Joonis 39. Ülesande 2 EV3 programmi näide.

I²C plokkis valida funktsiooniks “Write 8 byte”. Pöördenurga jagatis ühendada I²C ploki seitsmenda baidiga. Jagatise jääk, mis saadakse “Math” ploki “Advanced” funktsiooniga kasutades valemit “a%b”, ühendada I²C ploki kaheksanda baidiga. EV3 programmi fail asub lisana failis yl2.ev3s.

Arduino programmi koostamiseks avada ülesannete jaoks loodud programmi mall, mis asub lisana failis mall.ino ning muuta seda. Antud ülesande korral on mõistlik *pinMode()* määrata funktsioonis *setup()* mitte programmi tsüklis, sest väljundviik jääb programmi töötamise ajal samaks. Lugeses I²C siinilt andmeid, tuleb lugeda ainult kaks viimast baiti. Seega funktsiooni *andmedSisse()* *while*-tsükli tingimuseks kirjutada *Wire.available() > 2* ning tsükli

kehasse kirjutada `Wire.read()`; . Nii jääb lugemata ainult kaks viimast baiti. Need tuleb lugeda muutujatesse väljaspool `while`-tsükli. Näidislahendus asub lisana failis `yl2.ino`.

Tekkida võivad probleemid:

- Ühendus ei toimi korralikult, sest EV3 programmis on viivitus liiga väike.
Lahendus: suurendada viivitust.
- Valgusdiodiga ühendatud viigu number ei vasta Arduino programmis olevale viigu numbrile.
Lahendus: muuta programmis viigu number õigeks.
- “Addr” väärtus ei vasta Arduino programmis olevale “ALAMSEADME_ADDRESS” väärtusele.
Lahendus: seada „Addr“ väärtus ja “ALAMSEADME_ADDRESS” samaks.
- Valitud port ei vasta EV3 juhtploki pordile.
Lahendus: valida EV3 programmis õige port.

Ülesande alternatiivid:

- Kasutades RGB valgusdiodi ning lugedes mootori hetkevõimsust, muuta valgusdiodi värvust. Selleks saab kasutada näiteks Arduino funktsiooni `map()`. Arduino näidisprogramm on lisana failis `yl2_2.ino`.
- Leida Arduino funktsioonis `map()` kasutatav arv EV3 programmis selliselt, et vastus oleks väiksem kui 128. Nii ei pea saatma Arduinole mitut baiti üksteise järel, seega ei pea ka leidma jagatist ja jääki ning saab Arduinole saata ainult ühe baidi.

3.3 Ülesanne 3 - potentsiomeetriga EV3 suure mootori kiiruse muutmine

Tase: keskmine

Eesmärk: kasutades potentsiomeetrit, muuta EV3 mootori kiirust. Tutvuda Arduino funktsiooniga `analogRead()`.

Ülesande täitmiseks vajalikud vahendid:

- Punktis 2.4 toodud juhiste järgi ühendatud LEGO Mindstorms EV3 juhtplokk ja Arduino arendusplaat

- EV3 suur mootor
- Makettilaualle ühenduv potentsiomeeter (soovitavalt 10k)
- Makettilaud
- Makettilaua juhtmed
- Arvuti LEGO Mindstorms EV3 tarkvaraga, millesse imporditud firma Dexter Industries I²C plokk, ja Arduino arenduskeskkonnaga.
- Ülesannete lahendamiseks loodud Arduino programmi mall (Lisana failis mall.ino)

Ülesande püstitus:

Kontrollida EV3 mootori pöörlemise kiirust ja suunda kasutades potentsiomeetrit.

Lahenduse idee:

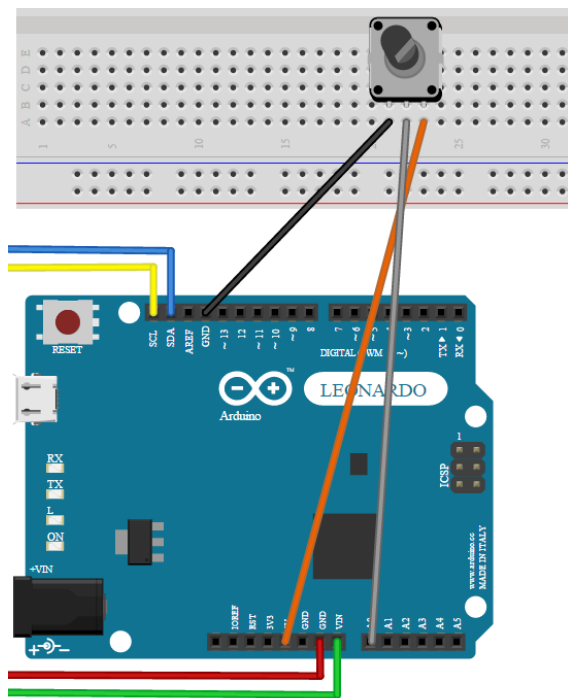
Ühendada potentsiomeeter makettilaua külge. Potentsiomeetril on kolm kontakti. Äärmised kontaktid on takisti kontaktid ja need tuleb ühendada Arduino arendusplaadi GND ja 5V viikudega. Keskmise kontakt on liuguri kontakt, mis tuleb ühendada Arduino arendusplaadi analoogviiguga. Nüüd, kui potentsiomeetrit pöörata, muutub keskmise kontakti väljundpinge vahemikus 0V-5V, olenevalt kummas suunas potentsiomeetrit pööratakse. Arduino arendusplaadi analoogviigu abil saab teisendada viigule rakendatava pinge 0V-5V täisarvuks 0-1023. Teiste sõnadega, potentsiomeetri nupu asukoha saab teisendada täisarvuks 0-1023 kasutades funktsiooni *analogRead(analoogViik)*.

EV3 mootori pöörlemise kiiruse kontrollimiseks kasutatakse arve vahemikus -100 kuni 100. Kuna negatiivset arvu on keeruline saata üle I²C siini, siis teisendada Arduino programmis loetud potentsiomeetri väärtus funktsiooni *map()* abil vahemikust 0-1023 vahemikku 0-200. Vahemiku laius jääb samaks, kuid arvud on kõik positiivsed. Saadud arv tuleb saata EV3 juhtploki. Millegipärast on EV3 programmis kasutatava Dexter Industries I²C ploki "AnalogRead" funktsioon ainuke, mis oskab lugeda saadetud baiti õigesti - see tähendab, et 8 biti kaupa, mitte 7 biti kaupa nagu ülejäänud funktsioonid. Sel põhjusel kasutada EV3 programmis just I²C ploki "AnalogRead" funktsiooni. Tagasi Arduino programmi juurde tulles - saadetav arv tuleb saata kahe baidi kaupa, sest I²C ploki "AnalogRead" funktsioon loeb esimese arvu, korrutab selle 256-ga ning liidab seejärel teise arvu juurde. Seega tuleb luua uus baidimassiiv *byte b[2];*, kus *b[0] = arv/256;* ning *b[1] = arv%256;*. Loodud massiivi saata EV3 juhtploki kasutades funktsiooni *Wire.write(b, 2);*.

EV3 programmis lugeda saadetud arv kasutades I²C plokki “AnalogRead” funktsiooni. Seejärel lahutada saadud arvust 100, sest arv peab asuma vahemikus -100 kuni 100, mitte 0 kuni 200. Vastus seada mootori juhtimise plokki “Power” argumendiks.

Üks võimalik lahendus:

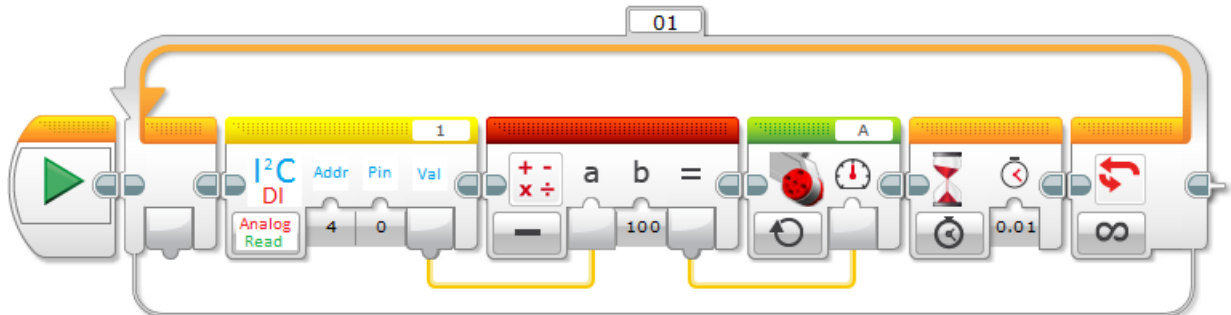
Ühendada potentsiomeeter makettlauda külge. Potentsiomeetri vasakpoolne kontakt ühendada Arduino arendusplaadi GND viigu külge ja parempoolne kontakt arendusplaadi 5V viigu külge. Keskmise kontakt ühendada arendusplaadi analoogviiguga A0. Ühendamise skeem on toodud joonisel 40.



Joonis 40. Potentsiomeetri ühendamine Arduino arendusplaadiga. Sinine, kollane, punane ja roheline juhe kuuluvad EV3 ühenduskaablisse, mis on ühendatud EV3 juhtploki.

Avada ülesannete jaoks loodud programmi mall, mis asub lisana failis mall.ino ning muuta seda. Programmi tsükklisse lisada `read int potVaartus = analogRead(0);` ja `int kiirus = map(potVaartus, 0, 1023, 0, 200);` ning saadetava arvu teisendamine kaheks baidiks - `byte b[2]; b[0] = arv/256;` ja `b[1] = arv%256;`. Funktsioonis `andmedValja()` kirjutada olemasoleva rea asemele `Wire.write(b, 2);`. Näidislahendus asub lisana failis yl3.ino.

Programm EV3 jaoks koosneb lõputust tsüklist, milles on I²C plokk “AnalogRead” funktsiooniga, “Math” plokk, “Large Motor” plokk ja viivituse plokk. Programm on kujutatud joonisel 41 ning asub lisana failis yl3.ev3s.



Joonis 41. Ülesande 4 EV3 programmi näide.

Tekkida võivad probleemid:

- Viivitus on liiga lühike ning mootor ei tööta korrektselt.
Lahendus: suurendada viivitust EV3 programmis.
- “Addr” väärtus ei vasta Arduino programmis olevale “ALAMSEADME_AADRESS” väärtusele.
Lahendus: seada „Addr“ väärtus ja „ALAMSEADME_AADRESS“ samaks.
- Valitud port ei vasta EV3 juhtploki pordile.
Lahendus: valida EV3 programmis õige port.

3.4 Ülesanne 4 - LEGO klotsidest ehitatud auto esirataste pööramine servomootoriga

Tase: keskmine

Eesmärk: ehitada auto, mille esirattaid saab pöörata servomootoriga. Tutvuda servomootori juhtimisega.

Ülesande täitmiseks vajalikud vahendid:

- Arduino arendusplaat
- Servomootor
- Makettlaua külge ühendatav potentsiomeeter (soovitavalt 10k Ω)
- Makettlaud

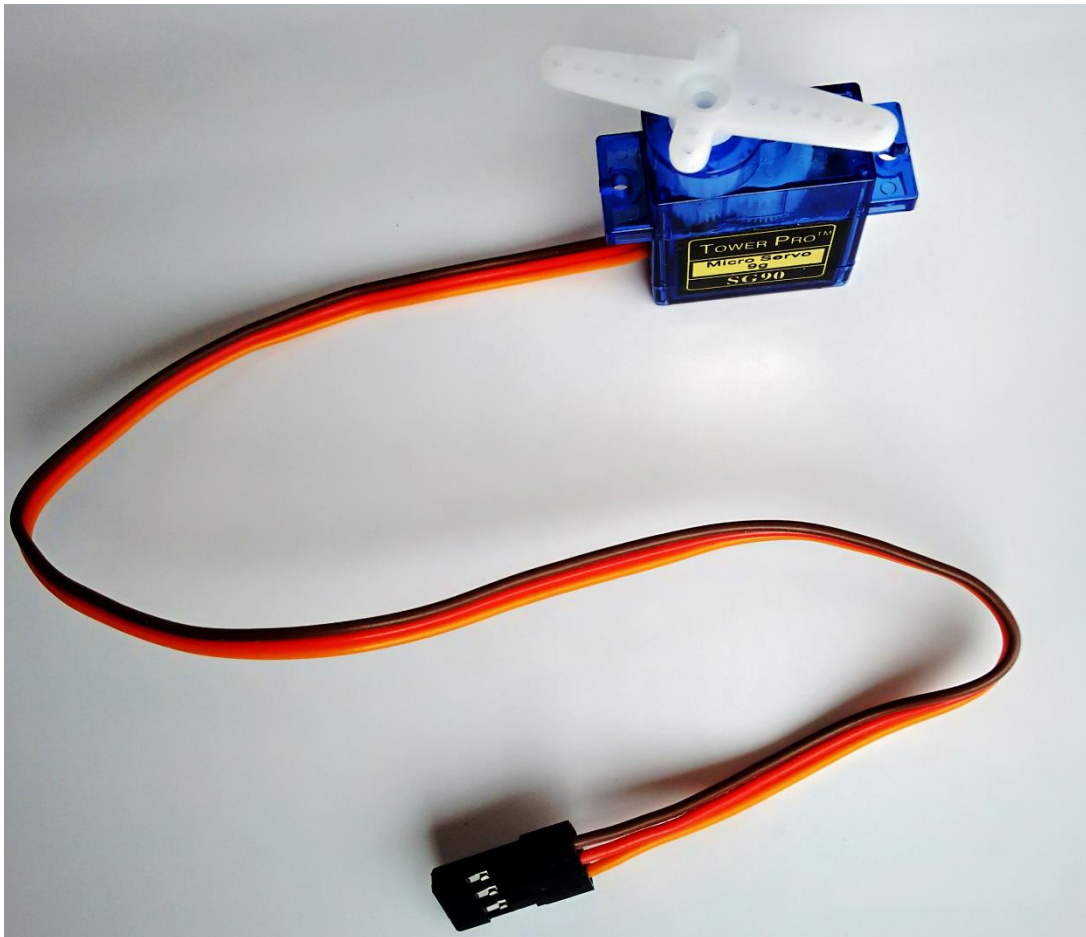
- Maketlatau ühendusjuhtmed
- LEGO klotsid ja rattad auto ehitamiseks

Ülesande püstitus:

Ehitada LEGO klotsidest auto, mille esirattaid saab servomootoriga juhtida. Servomootori kontrollimiseks kasutada potentsiomeetrit. Antud ülesandes EV3 juhtplokki vaja ei lähe.

Lahenduse idee:

Ehitada auto mille esirattaid saab paralleelselt pöörata. Rataste pööramist kontrollida servomootoriga. Servomootor, (Joonis 42) on väike mootor, mille hoova asendit saab väga täpselt kontrollida. Servomootori liikuv hoob saab hoida kindlat asendit vahemikus 0-180 kraadi. Seega on servomootorit erinevate asjade täpseks liigutamiseks väga mugav kasutada.



Joonis 42. TowerPro SG90 servomootor.

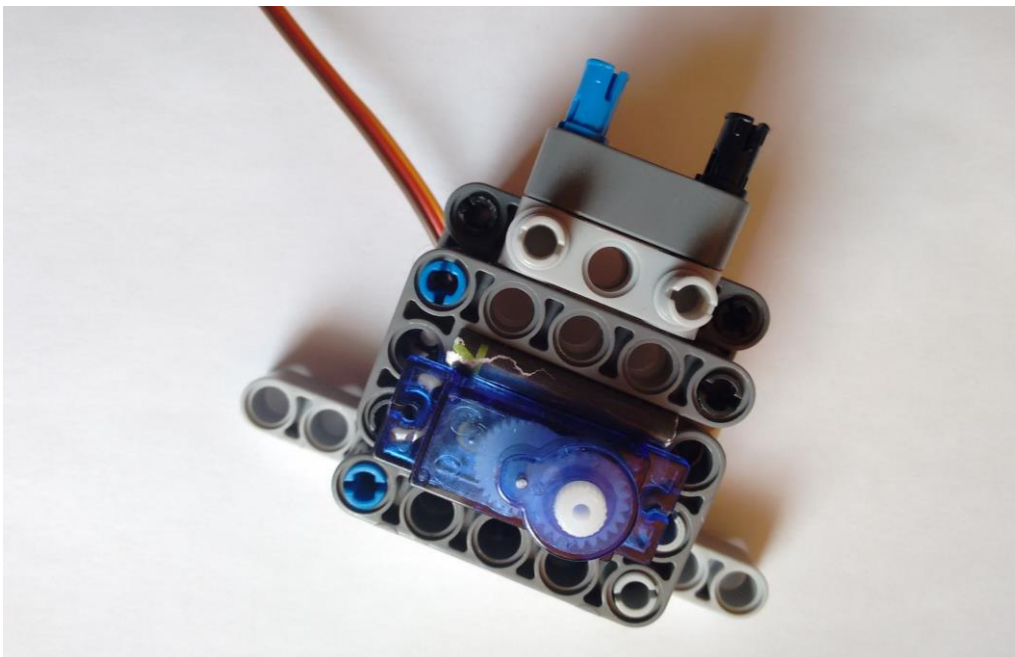
Servomootori juhtmed tuleb ühendada Arduino arendusplaadi külge. Joonisel 42 oleva servomootori juhtmete värvused tähendavad järgmist: pruun – maandus (GND), punane – toide

(VCC) ja oranž - signaal. Need tuleb ühendada Arduino arendusplaadi vastavate viikudega. Lisaks tuleb arendusplaadiga ühendada potentsiomeeter.

Arduino programmis tuleb kasutada teeki Servo (`#include <Servo.h>`) ning luua uus *Servo* objekt. Seejärel määrata viik, mille külge servomootor on ühendatud. Lugeda potentsiomeetri väärtus ja teisendada see vahemikku 1-180 kasutades funktsiooni *map()*. Teisendatud väärtus ongi nurk, mis määrab servomootori asendi ning tuleb seega edastada servomootorile.

Üks võimalik lahendus:

Ehitada LEGO klotsidest lihtne kere, mille külge saab kinnitada tagarattad ja paralleelselt pööratavad esirattad. Seejärel kinnitada kere külge servomootor. Selleks ehitada servomootori ümber LEGO klotsidest raam, mille abil servomootori ülejäänud kere külge saab kinnitada (Joonis 43).



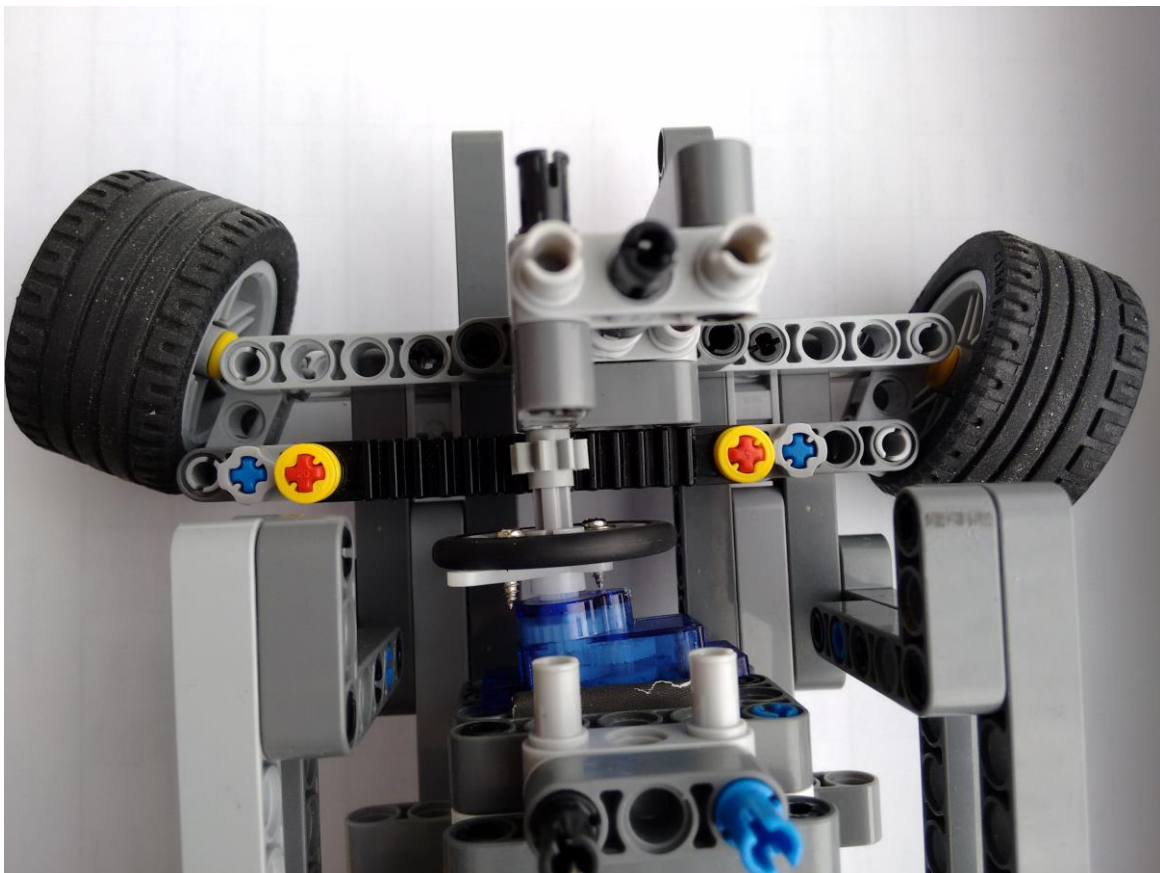
Joonis 43. Servomootori ümber ehitatud raam servomootori kinnitamiseks LEGO klotside külge.

Servomootori hoovaga on vaja ühendada LEGO hammasratas. Selle jaoks saab näiteks meisterdada ülemineku (Joonis 44) kasutades kitsast LEGO ratast ja servomootoriga kaasatulevaid kruve.



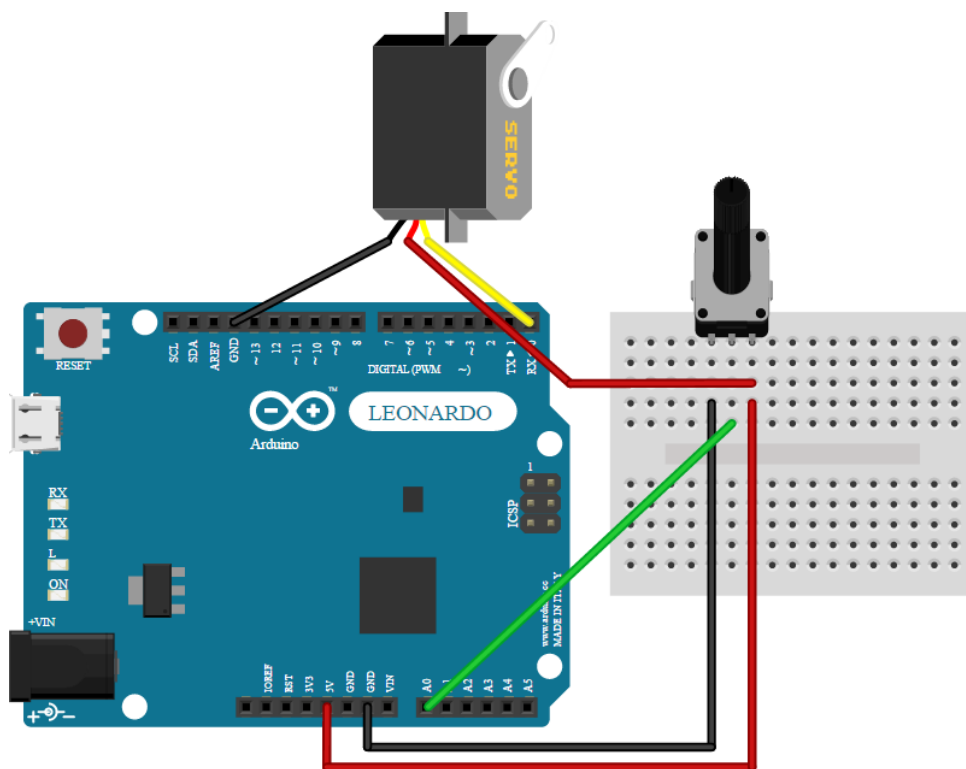
Joonis 44. Servomootori hoova üleminek LEGO hammasrattale.

Joonisel 44 toodud adapteri üks ots tuleb kinnitada servomootori külge ja teine ots esirattaid juhtiva klotsi küljes olevale hammaslatile. Juhtimismehhanism on kujutatud joonisel 45.



Joonis 45. Valminud auto esirataste juhtimissüsteem.

Servomootor ja potentsiomeeter tuleb ühendada Arduino arendusplaadi külge nii nagu on näidatud joonisel 46.



Joonis 46. Servomootori ja potentsiomeetri ühendamine Aduino arendusplaadi külge.

Arduino programmis tuleb lisada teek `Servo` (`#include <Servo.h>`) ja luua uus `Servo` objekt (`Servo minuServo;`). Funktsioonis `setup()` tuleb määrata viik, mille külge on servomootori signaaljuhe ühendatud - `minuServo.attach(0);`. Funktsioonis `loop()` tuleb lugeda potentsiomeetri väärtus kasutades funktsiooni `analogRead(A0);`. Loetud väärtus teisendada funktsiooni `map()` abil vahemikust 0-1023 vahemikku 1-180. Seejärel tuleb edastada teisendatud väärtus ehk nurk servomootorile, kasutades meetodit `minuServo.write(nurk);`. Näidislahendus asub lisana failis `yl4.ino`.

Tekkida võivad probleemid:

- Potentsiomeeter on ühendatud valede viikude külge
Lahendus: ühendada potentsiomeeter joonisel 45 kujutatud skeemi aslusel.
- Programm ei kompilleeru, sest pole lisatud teeki `Servo`
Lahendus: lisada programmi agusesse rida `#include <Servo.h>`.

3.5 Ülesanne 5 - puldist juhitud auto, millel on esi ja tagatuled

Tase: raske

Eesmärk: ehitada LEGO klotsidest auto, mida saab puldist juhtida. Eelnevate ülesannete kombineerimine.

Ülesande täitmiseks vajalikud vahendid:

- Punktis 2.4 toodud juhiste järgi ühendatud LEGO Mindstorms EV3 juhtplokk ja Arduino arendusplaat
- EV3 suur mootor
- Kolm valget valgusdiodi
- Neli oranži valgusdiodi
- Kaks punast valgusdiodi
- Üheksa 150Ω takistit
- Kaks makettlauda külge ühendatavat potentsiomeeterit (soovitavalt 10kΩ)
- Kolm makettlauda külge ühendatavat mikronuppu
- Servomootor
- Kaks makettlauda
- Makettlauda ühendusjuhtmed
- Seitse juhet pikkusega ~1m
- Arvuti LEGO Mindstorms EV3 tarkvaraga, millesse imporditud firma Dexter Industries I²C plokk, ja Arduino arenduskeskkonnaga.
- Ülesannete lahendamiseks loodud Arduino programmi mall (Lisana failis mall.ino)

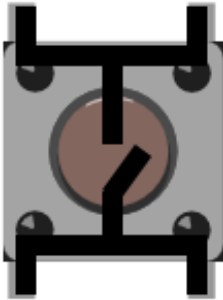
Ülesande püstitus:

Ehitada puldist juhitud auto, millel on esi-, taga- ja suunatud ning tagurdustuli. Esirattaid juhtida servomootori abil.

Lahenduse idee:

Ühendada eelmistest ülesannetest saadud teadmised ja kombineerida need, et ehitada puldist juhitud LEGO auto. Valgusdiodide, servomootori ja potentsiomeetri tööpõhimõtet kirjeldati eelmistes ülesannetes. Lisaks eelnevalt kasutatud riistvarale tuleb selles ülesandes kasutada ka mikronuppe. Mikronuppe kasutada antud ülesandes tulede lülitamiseks.

Mikronupul on neli jalga - kummagi poole kaks jalga on omavahel ühenduses. Nupu alla vajutamine ühendab kummagi poole jalad omavahel (Joonis 47).



Joonis 47. Mikronupu skeem.

Kui mikronupu üks pool ühendada Arduino arendusplaadi GND viiguga ja teine pool mõne digitaalviiguga, siis on võimalik nupuvajutust registreerida. Selleks tuleb programmis funktsiooni *pinMode(viik, olek)* argumendi *olek* väärtuseks seada “INPUT_PULLUP”, mis rakendab viigule läbi takisti pinge 5V. Arduino programmis tagastab nüüd funktsioon *digitalRead(viik)*; väärtuse “HIGH”, sest viigule rakendub pinge 5V. Nupu vajutamisega rakendub viigule aga 0V, ning funktsioon *digitalRead(viik)*; tagastab väärtuse “LOW”. Nii saabki nupuvajutust registreerida.

LEGO klotsidest tuleb ehitada nelja rattaga auto, mille tagumisi rattaid veab EV3 suur mootor. Mootori pöörlemise kiirust kontrollida potentsiomeetri abil. Esiratuste juhtimise süsteem tuleb ehitada samamoodi nagu on kirjeldatud ülesandes 4. Auto külge tuleb kinnitada esi-, taga- ning suunatud ja tagurdustuli ehk siis neli oranži, kaks punast ja kolm valget valgusdiodi.

Iga valgusdiod on vaja arendusplaadi külge ühendada 150Ω takisti abil. Parema suunatudle valgusdiodid ühendada sama väljundviiguga. Sama teha ka vasaku suunatudle, tagatudle ning esitudede valgusdiodidega. Tagurdustule valgusdiodid ühendada samuti väljundviiguga. Kokku läheb valgusdiodide ühendamise jaoks vaja viit väljundviiku.

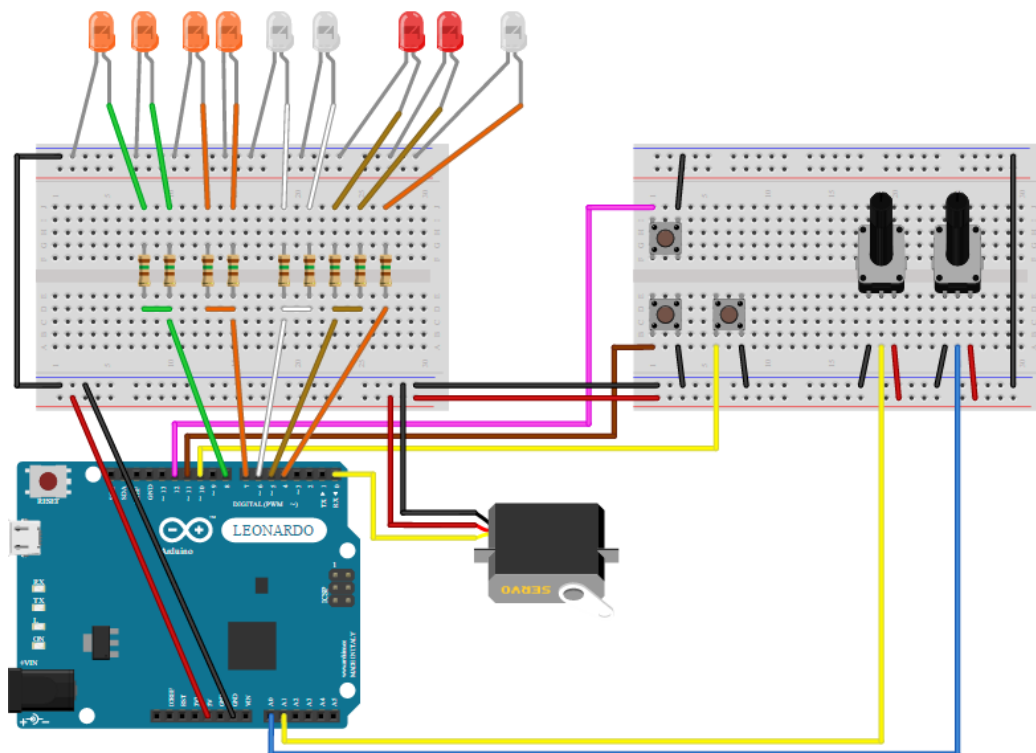
Servomootori ja EV3 suure mootori juhtimiseks on vaja kasutada potentsiomeetreid. Potentsiomeetrid tuleb ühendada ühe maketlauri külge, sama maketlauri külge tuleb ühendada ka kolm mikronuppu. Potentsiomeetrite äärmised kontaktid tuleb ühendada 5V liiniga ja maanduse liiniga, keskmised kontaktid aga eraldi Arduino arendusplaadi analoogsisendi viikudega. Mikronupud tuleb ühendada ühelt poolt maandusega ning teiselt poolt iga nupp eraldi Arduino arendusplaadi digitaalviiguga. Seega saab ühest maketlaurist pult, millega autot ja tulesid juhtida. Puldiga on vaja ühendada seitse juhet - maandus, 5V, kaks potentsiomeetri

väljundit ja kolm mikronupu väljundit. Need juhtmed peaksid olema pikemad, et oleks mugav autot juhtida.

Arduino programmis ei saa kasutada funktsiooni *delay(aeg)* valgusdiodide vilgutamiseks, sest siis jääks seisma ka kogu ülejäänud programmi töö. Valgusdiodide vilgutamiseks tuleb funktsiooni *millis()* abil võrrelda kas eelmise valgusdiodi vilkumise ajast on möödunud sobiv aeg ning vastavalt siis valgusdiodi sisse või välja lülitama.

Üks võimalik lahendus:

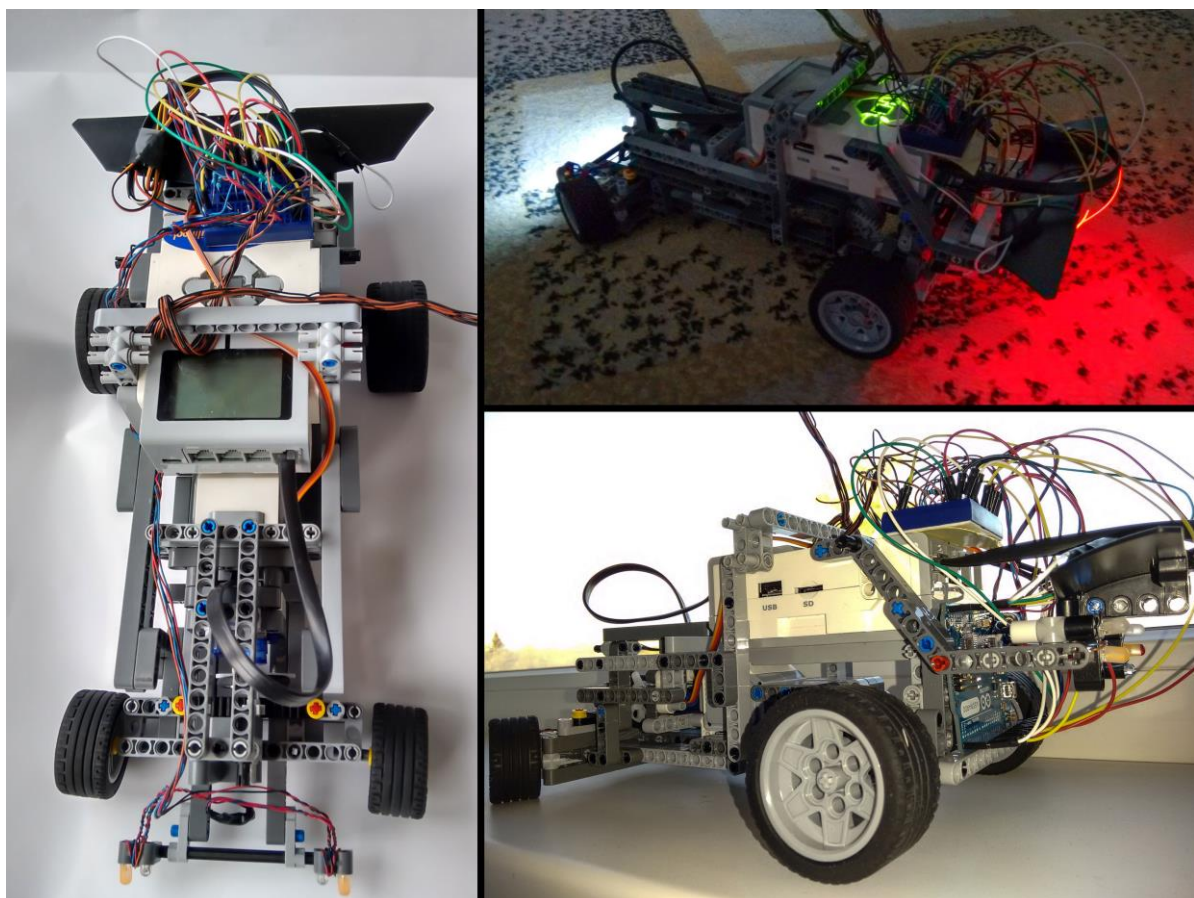
Ehitada LEGO klotsidest nelja rattaga auto, mille tagumisi rattaid veab EV3 suur mootor ning esirattaid pöörab ülesandes 4 toodud juhtimissüsteemi põhjal servomootor. Auto esiküljele kinnitada kaks valget valgusdiodi, millest saavad esituled ning kaks oranži valgusdiodi, millest saavad parem ja vasak suunatuli. Auto tagaküljele kinnitada kaks punast valgusdiodi, millest saavad tagatuled ja üks valge valgusdiod, millest saab tagurdustuli. Lisaks tuleb auto külge kinnitada LEGO klotside või kahepoolse teibi abil Arduino arendusplaat. Valgusdiodid ühendada takistite kaudu Arduino arendusplaadiga. Samuti ühendada eraldi makettlauale kaks potentsiomeetrit ning kolm mikronappu. Kõikide komponentide ühendamise skeem on toodud joonisel 48. Lisaks tuleb Arduino arendusplaadiga ühendada EV3 juhtploki tulevad kollane, sinine, roheline ja punane juhe.



Joonis 48. Ülesande 5 jaoks vajaminevate komponentide ühendamine Arduino arendusplaadiga.

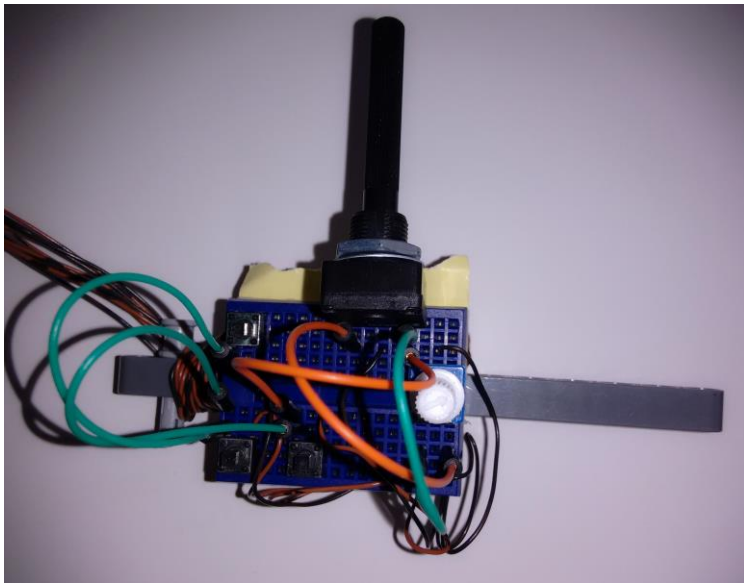
EV3 programm on sama, mis ülesandes 3 toodud ning asub lisana failis yl3.ev3s. Arduino programmi koostamiseks avada ülesannete jaoks loodud programmi mall, mis asub lisana failis mall.ino ning muuta seda. Funktsiooni *setup()* lisada korrektsed viikude seadistused (*pinMode(viik, seadistus)*) – mikronuppudega ühenduses olevate viikude seadistuseks seada “INPUT_PULLUP” ning valgusdiodidega ühenduses olevate viikude seadistuseks seada “OUPUT”. Suure mootori juhtimise ja esirataste pööramise koodi leiab eelmistest ülesannetest. Järgmisena on vaja lisada muutujaid mõlema poole suunatulede jaoks, et saaks määrata kas suunatuli on sisse lülitatud või mitte. Samuti on muutujaid vaja valgusdiodide oleku jaoks. Lisada on vaja ka muutujad, kus hoitakse aega, millal mikronuppu on vajutatud. Seda aega võrreldakse iga vajutuse ajaga ning ainult siis, kui piisav aeg on möödunud eelmisest nupuvajutusest, registreeritakse uus nupuvajutus. Ilma nende muutujateta registreeritakse iga nupuvajutuse korral mitukümmend korda nupuvajutus ning ei saa tule täpset olekut (sees/väljas) määrata. Kogu lahendus asub lisana failis yl5.ino.

Valminud auto on kujutatud joonisel 49.



Joonis 49. Ülesande 5 põhjal valminud puldist juhitud LEGO auto.

Auto juhtimiseks vajalik pult on kujutatud joonisel 50.



Joonis 50. Ülesande 5 põhjal valminud LEGO auto juhtimiseks vajalik pult.

Tekkida võivad probleemid:

- Valgusdiodid on ühendatud valede viikude külge
Lahendus: kontrollida ühendused üle ning viga parandada
- “Addr” väärtus ei vasta Arduino programmis olevale “ALAMSEADME_ADDRESS” väärtusele.
Lahendus: seada „Addr“ väärtus ja „ALAMSEADME_ADDRESS” samaks.
- Valitud port ei vasta EV3 juhtploki pordile.
Lahendus: valida EV3 programmis õige port.

Ülesande alternatiivid:

- Juhtida valgusdiodide tööd EV3 programmiga.
- Lisada tagumistele punastele tuledle pidurdamise funktsioon ehk kui auto kiirus väheneb, siis lähevad punased tuled heledamalt põlema.
- Muuta auto isesõitvaks.

Käesolevas peatükis anti viis suureneva raskusastmega ülesannet LEGO Mindstorms EV3 ja Arduino arendusplaadi kooskasutamiseks. Iga ülesande juures olevad alternatiivsed ülesanded pakuvad lisaks põhiülesannete lahendamisele lisa väljakutset ja enese proovile panekut Arduino arendusplaadi ja EV3 juhtploki programmeerimisel ja nende koos kasutamisel.

Kokkuvõte

Antud bakalaureuse töö eesmärgiks oli luua eestikeelne õppematerjal LEGO Mindstorms EV3 juhtploki ja Arduino arendusplaadi ühendamiseks ja kooskasutamiseks. Valminud õppematerjali põhjal saab hõlpsalt eelnimetatud seadmed omavahel ühendada ning neid koos kasutada. Lisaks võimaldavad toodud ülesanded koos lahenduste ja selgitustega tutvuda Arduino arendusplaadi ja EV3 juhtploki kasutamisega ning Arduino arendusplaadi programmeerimisega

Valminud õppematerjali saab kasutada erinevates reaalainetega seonduvates koolitundides ning robotikaringides. Materjal on sobiv gümnaasiumiastme õpilastele ning EV3 kasutamise ja programmeerimisega toime tulevatele õpilastele. Lisaks pakub õppematerjal võimalust sujuvalt üle minna EV3 programmeerimiseks kasutatavalt visuaalselt programmeerimiskeelelt Arduino programmeerimiseks kasutatavale objektorienteeritud programmeerimiskeelele.

Lõputöö kirjutamine andis autorile uusi kogemusi robotika valdkonnas. Huvitav oli programmeerida kahte erinevat seadet koos töötama, kui ühe seadme programmeerimiseks kasutati visuaalset programmeerimiskeelt ning teise korral objektorienteeritud programmeerimiskeelt. Töö lõplik maht kujunes oodatust natuke suuremaks, kuid tänu sellele on töös palju selgitusi, mis aitavad materjalist paremini aru saada.

Kasutatud kirjandus

- [1] „Google Trends,“ Google, [Võrgumaterjal] <http://www.google.com/trends/explore#q=arduino%2C%20raspberry%20pi%2C%20teensy%2C%20msp430&cmpt=q> (kasutatud 23.04.2015).
- [2] „Arduino Leonardo,“ Arduino, [Võrgumaterjal]. <http://arduino.cc/en/Main/ArduinoBoardLeonardo> (kasutatud 25.04.2015).
- [3] „Teensy 3.1,“ PJRC, [Võrgumaterjal]. <https://www.pjrc.com/teensy/teensy31.html> (kasutatud 12.05.2015).
- [4] „MSP430 Launchpad,“ Texas Instruments, [Võrgumaterjal]. <http://www.ti.com/ww/en/launchpad/launchpads-msp430-msp-exp430g2.html> (kasutatud 25.04.2015).
- [5] „Raspberry Pi,“ Wikipedia, [Võrgumaterjal]. http://en.wikipedia.org/wiki/Raspberry_Pi (kasutatud 12.05.2015).
- [6] A. Peets, „Methods and tools used in fast development of robotics projects,“ *REVISTA EDUCACIÓN EN INGENIERÍA*, pp. 140-143, 15.12.2013.
- [7] „Introduction,“ Arduino, [Võrgumaterjal]. <http://arduino.cc/en/guide/introduction> (kasutatud 25.04.2015).
- [8] „Arduino,“ Wikipedia, [Võrgumaterjal]. <http://en.wikipedia.org/wiki/Arduino> (kasutatud 11.05.2015).
- [9] „Arduino_Uno_-_R3.jpg,“ Wikimedia, [Võrgumaterjal]. http://upload.wikimedia.org/wikipedia/commons/3/38/Arduino_Uno_-_R3.jpg (kasutatud 15.03.2015).
- [10] „USBSerial.jpg,“ Arduino, [Võrgumaterjal]. <http://www.arduino.cc/en/uploads/Main/USBSerial.jpg> (kasutatud 10.04.2015).
- [11] „Compare board specs,“ Arduino, [Võrgumaterjal]. <http://arduino.cc/en/Products.Compare> (kasutatud 11.05.2015).
- [12] „ArduinoLeonardoFront_2.jpg,“ Arduino, [Võrgumaterjal]. http://arduino.cc/en/uploads/Main/ArduinoLeonardoFront_2.jpg (kasutatud 15.03.2015).
- [13] „Arduino Uno,“ Arduino, [Võrgumaterjal]. <http://arduino.cc/en/Main/ArduinoBoardUno> (kasutatud 25.04.2015).

- [14] „Arduino Mega,“ Arduino, [Võrgumaterjal]. <http://arduino.cc/en/Main/ArduinoBoardMega> (kasutatud 25.04.2015).
- [15] „LilyPad Arduino,“ Arduino, [Võrgumaterjal]. <http://arduino.cc/en/Main/ArduinoBoardLilyPadSimple> (kasutatud 25.04.2015).
- [16] „Arduino Yun,“ Arduino, [Võrgumaterjal]. <http://arduino.cc/en/Main/ArduinoBoardYun> (kasutatud 25.04.2015).
- [17] „CONNECTING THE EV3 AND THE ARDUINO,“ Dexter Industries, [Võrgumaterjal]. <http://www.dexterindustries.com/howto/connecting-ev3-arduino/> (kasutatud 12.05.2015).
- [18] „Lego_mindstorms_nxt_cable.jpg,“ Wikimedia, [Võrgumaterjal]. http://upload.wikimedia.org/wikipedia/commons/c/ce/Lego_mindstorms_nxt_cable.jpg (kasutatud 31.03.2015).
- [19] „Raspi.png,“ UT, [Võrgumaterjal]. https://courses.cs.ut.ee/topic_wikis/kids/uploads/Main/Raspi.png (kasutatud 12.05.2015).

Lisad

I. Arduino arendusplaatide mudelite riistvaraline võrdlus

Tabel 2. Arduino arendusplaatide mudelite riistvaraline võrdlus [11].

Nimi	Protsessor	Tööpinge/ Sisend- pinge	CPU Kiirus	Analoog Sisend/ Väljund	Digitaal IO/PWM	EEP- ROM [KB]	SRAM [KB]	Flash [KB]	USB	UART
Uno	ATmega328	5 V/7-12 V	16MHz	6/0	14/6	1	2	32	Regular	1
Due	AT91SAM3X8E	3.3 V/7-12 V	84MHz	12/2	54/12	-	96	512	2 Micro	4
Leonardo	ATmega32u4	5 V/7-12 V	16MHz	12/0	20/7	1	2.5	32	Micro	1
Mega 2560	ATmega2560	5 V/7-12 V	16MHz	16/0	54/15	4	8	256	Regular	4
Mega ADK	ATmega2560	5 V/7-12 V	16MHz	16/0	54/15	4	8	256	Regular	4
Micro	ATmega32u4	5 V/7-12 V	16MHz	12/0	20/7	1	2.5	32	Micro	1
Mini	ATmega328	5 V/7-9 V	16MHz	8/0	14/6	1	2	32	-	-
Nano	ATmega168	5 V/7-9 V	16MHz	8/0	14/6	0.512	1	16	Mini-B	1
	ATmega328					1	2	32		
Ethernet	ATmega328	5 V/7-12 V	16MHz	6/0	14/4	1	2	32	Regular	-
Esplora	ATmega32u4	5 V/7-12 V	16MHz	-	-	1	2.5	32	Micro	-
ArduinoBT	ATmega328	5 V/2.5-12 V	16MHz	6/0	14/6	1	2	32	-	1
Fio	ATmega328P	3.3 V/3.7-7 V	8MHz	8/0	14/6	1	2	32	Mini	1
Pro (168)	ATmega168	3.3 V/3.35-12 V	8MHz	6/0	14/6	0.512	1	16	-	1
Pro (328)	ATmega328	5 V/5-12 V	16MHz	6/0	14/6	1	2	32	-	1
Pro Mini	ATmega168	3.3 V/3.35-12 V	8MHz	6/0	14/6	0.512	1	16	-	1
		5 V/5-12 V	16MHz							
LilyPad	ATmega168V	2.7-5.5 V/2.7-5.5 V	8MHz	6/0	14/6	0.512	1	16	-	-
	ATmega328V									
LilyPad USB	ATmega32u4	3.3 V/3.8-5V	8MHz	4/0	9/4	1	2.5	32	Micro	-
LilyPad Simple	ATmega328	2.7-5.5 V/2.7-5.5 V	8MHz	4/0	9/4	1	2	32	-	-
LilyPad Simple-Snap										
Yun	ATmega32u4	5 V	16MHz	12/0	20/7	1	2.5	32	Micro	1

II. Töö lisana kaasa tulevad failid

Tabelis 3 on toodud punktis 2.6 kirjeldatud ühenduse testimise jaoks vajalikud failid ning peatükis 3 antud ülesannetes nimetatud Arduino programmi mall ja ülesannete lahenduste failid.

Tabel 3. Töö lisasse kuuluvad failid.

Faili kirjeldus	Faili nimi
Ühenduse testimine (Arduino)	test1.ino
Ühenduse testimine (EV3)	test1.ev3s
Ülesannete jaoks loodud Arduino programmi mall	mall.ino
Ülesanne 1 lahendus (Arduino)	yl1.ino
Ülesande 1 lahendus (EV3)	yl1.ev3s
Ülesande 2 lahendus (Arduino)	yl2.ino
Ülesande 2 lahendus (EV3)	yl2.ev3s
Ülesande 2 lõpus toodud alternatiivse ülesande lahendus	yl2_2.ino
Ülesande 3 lahendus (Arduino)	yl3.ino
Ülesande 3 lahendus (EV3)	yl3.ev3s
Ülesande 4 lahendus (Arduino)	yl4.ino
Ülesande 5 lahendus (Arduino)	yl5.ino

III. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina **Lauri Uusmaa** (sünnikuupäev: 11.06.1993)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose **Arduino kasutamine koos LEGO Mindstorms EV3 robotiga**, mille juhendajad on Anne Villems, 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, **14.05.2015**