



TALLINNA TEHNIKAÜLIKOOL  
INSENERITEADUSKOND  
Virumaa kolledž

## **Programmeeritav taevakaart**

### **Programmable SkyMap**

Rakendusinfotehnoloogia ÕPPEKAVA LÕPUTÖÖ

Üliõpilane: Margit Mark

Üliõpilaskood: 124438RDIR

Juhendaja: Sergei Ponomar, lektor

# AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

“29” mai 2021.

Autor: Margit Mark

/ digitaalselt allkirjastatud /

Töö vastab rakenduskõrgharidusõppe lõputööle/magistritööle esitatud nõuetele  
“29” mai 2021.

Juhendaja: Sergei Ponomar

/ digitaalselt allkirjastatud /

Kaitsmisele lubatud

“29” mai 2021.

Kaitsmiskomisjoni esimees Mare Roosileht /digitaalselt allkirjastatud/

/ nimi ja allkiri /

## **LIHTLITSENTS LÕPUTÖÖ ÜLDSUSELE KÄTTESAADAVAKS TEGEMISEKS JA REPRODUTSEERIMISEKS**

Mina Margit Mark (sünnikuupäev: 21.04.1988)

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose Programmeeritav taevakaart, mille juhendaja on Sergei Ponomar,
  - 1.1. reprodutseerimiseks säilitamise ja elektroonilise avaldamise eesmärgil, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu 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 kolmandate isikute intellektuaalomandi ega isikuandmete kaitse seadusest ja teistest õigusaktidest tulenevaid õigusi.

# TalTech Inseneriteaduskond Virumaa kolledž

## LÕPUTÖÖ ÜLESANNE

**Üliõpilane:** Margit Mark, 124438RDIR

Õppekava, peaariala: RDIR02/09 Rakendusinfotehnoloogia

Juhendaja(d): lektor, Sergei Ponomar, sergei.ponomar@taltech.ee

Konsultant: nimi, amet

ettevõtte, telefon, e-post

### Lõputöö teema:

(eesti keeles) Programmeeritav taevakaart

(inglise keeles) Programmable SkyMap

### Lõputöö põhieesmärgid:

1. Luua valguslahendusega programmeeritav taevakaart
2. Luua õppevahend, millega on lihtne õppida taevatähti
- 3.

### Lõputöö etapid ja ajakava:

Nr	Ülesande kirjeldus	Tähtaeg
1.	Uurida prototüübi valmistamiseks vajalike komponentide alast kirjandust	Märts 2021
2.	Võrrelda erinevaid arendusplatvorme	Märts 2021
3.	Sobivate taevakehade valimine ning luua vastav kaart	Aprill 2021
4.	Valmistada taevakaardi valguspaneel	Aprill 2021
5.	Töötada välja juhtpaneeli skeem	Mai 2021
6.	Ühendada valgustus elektriskeemi	Mai 2021
7.	Töötada välja programm	Mai 2021

**Töö keel:** eesti

**Lõputöö esitamise tähtaeg:** "29"mai 2021a

**Üliõpilane:** Margit Mark /digitaalselt allkirjastatud/  
/allkiri/

"29"mai 2021a

**Juhendaja:** Sergei Ponomar /digitaalselt allkirjastatud/  
/allkiri/

“29”mai 2021a

**Konsultant:** .....  
/allkiri/

“.....”..... 20.....a

**Programmijuht:** Žanna Gratšjova /digitaalselt allkirjastatud/  
/allkiri/

“02”juuni 2021a

# SISUKORD

EESSÖNA .....	7
LÜHENDITE JA TÄHISTE LOETELU .....	8
SISSEJUHATUS .....	9
1 PROGRAMMEERITAVA TAEVAKAARDI LOOMISEKS VAJALIK TEOORIA ÜLEVAADE	11
1.1. Trükkplaadil kasutatavad komponendid .....	11
1.1.1 Takisti.....	11
1.1.2 Diod.....	11
1.1.3 Transistor.....	12
1.1.4 ATMEGA328P-PU .....	12
1.1.5 TLC5940/TLC5940NT .....	12
1.2. Arendusplatvormide võrdlus .....	12
1.2.1 Arduino UNO ja Arduino Nano .....	13
1.2.2 Teensy .....	13
1.2.3 Raspberry PI.....	13
1.3. Arduino .....	14
1.4. Arduino keel .....	15
1.5. Taevakaardi mudelid.....	16
2 PROGRAMMEERITAVA TAEVAKAARDI KONSTRUEERIMINE .....	17
2.1. Juhtpaneel.....	17
2.2. Taevakaardi valguspaneel .....	23
2.3. Juhtpaneeli programmeerimine .....	28
2.4. Juhtpaneeli valmistamise probleemid .....	30
KOKKUVÕTE .....	33
SUMMARY.....	34
KASUTATUD KIRJANDUSE LOETELU .....	35
LISAD .....	37
LISA 1 Juhtpaneeli valmistamiseks vajalikud komponendid ning nende antud projektis olnud maksumus: .....	37
LISA 2 Programmikood.....	38

## EESSÕNA

Lõputöö teema sõnastati töö autori Margit Mark algatusel eesmärgiga luua taevakaardil baseeruv valguspaneel, mis oma õdusa valguslahendusega on interjööri element. Eelkõige soovib autor antud lõputööd kasutada kui õppematerjalina taevatähtede õppimisel ning juhendmaterjalina järgmiste valguspaneelide valmistamisel.

Töö esimeses peatükis käsitletakse programmeeritava taevakaardi loomiseks kasutatavate komponentide tööpõhimõtteid ja teemakohast kirjandust. Teises peatükis kirjutatakse detailsemalt programmeeritava taevakaardi erinevate osade valmistamisest. Põhiliselt on antud lõputöö teoreetilise osa info ja andmete leidmiseks kasutatud internetti, kuid on ka kasutatud raamatuid ja ajakirju raamatukogust.

Programmeeritava taevakaardi lõputöö loomisel abistas TalTech Virumaa Kolledži poolne juhendaja Sergei Ponomar, inspireeris õde Kristi Talli ning innustasid pereliikmed. Aitäh Teile!

Rakenduskõrgharidusõppe lõputöö „Programmeeritav taevakaart“ ideaalne õppevahend tähtkujude ja tähtede õppimiseks. Selles lõputöös on võimalik tutvuda taevakaardi konstrueerimiseks vajalike komponentidega ja vaadata Arduino Nanole loodud C-keeles kirjutatud programmikoodi.

Lõputöö märksõnad on: programmeeritav taevakaart, taevatähtede õppimine, valguskuidudega valguspaneel, Arduino Nano

## LÜHENDITE JA TÄHISTE LOETELU

1. °C – Celsiuse skaalal temperatuur
2. C-keel – C programmeerimise keel
3. C++ - C++ programmeerimise keel
4. DMX - digital multiplex kontrolleriid
5. kOhm – kilo oom
6. LED - valgusdiod (ingl. light emitting diode)
7. NPN- ja PNP-transistorid – bipolaartransistor, mis võib olla n- või p-juhtitavusega
8. mA - megaamper
9. MHz – megahertz
10. RGB – punane, roheline, sinine (ingl. red, green, blue)
11. SD-mälukaart - Secure Digital väikmäluga mälukaart
12. V - volt



## SISSEJUHATUS

Tänapäeval kasutatakse väga palju traditsiooniliste hõõglampide asemel LEDe (light emitting diode, valgusdiod), kuna need on võimekamad, energiasäästlikumad ning punktuaalsemad. Antud lõputöö autor on otsustanud ehitada LEDide juhtsüsteemi (kontrolleri), mille abil on võimalik juhtida suurt kogust LEDe. Lõputöö eesmärk on ehitada kontroller tasapinnalise taevakaardi valguspaneelile, millel asuvad valgustatud tähtkujud, mida on võimalik ükshaaval sisse/välja lülitada. Lõputöö peamisteks komponentideks on LED paneel ja sellel asuvad LEDid, mida juhitakse kontrolleriga. Need kaks osa tööst on muutumatud ning neid võib paigaldada ükskõik kuhu. Kogu süsteemi rakendamiseks ühendatakse LED paneel valguskaablitega taevakaardi paneeli külge.

LEDide poolt kiirgava valguse ning taevakaardi projitseerimiseks ekraanpinnale kasutatakse kiudoptikat ning taevakaardi tähtkujude esitlemiseks ehitatakse teisaldatav plaat mõõtmetega 1525 mm x 1040 mm. Seega, võib väita, et antud töös pöörab autor tähelepanu ehisvalgustusele ning elektroonikale, mis valgusti süsteemselt tööle paneb. Loodud on palju keerukaid ja kalleid süsteeme (DMX - digital multiplex kontrollerid, Intelligentsed LED kontrollerid). Suur osa neist süsteemidest on piiratud kindlate programmidega ning kasutajal ei ole võimalik neid oma soovi kohaselt ümber programmeerida.

Poodides müüdadavad valguse juhtsüsteemid on suutelised küll valgusallikaid vilgutama, kuid kasutaja jaoks ei ole selline lahendus piisavalt paindlik. Selle tõttu tuleb seadme konstrueerimisel arvestada täiustustega:

1. programmeeritavus (iga valgusallikas peab olema eraldi sisse ning välja lülitatav)
2. minimalistlikud dimensioonid (seadme kiibistik peab olema väike ning kompaktne)

Juhtseadme süsteemi täiustades tuleb leida võimalikult lihtsasti programmeeritavad ning omavahel sobituvad materjalid, mis alluksid programmikäsklustele ning ühilduksid Arduinoga. Samuti tuleb läbi mõelda, kuidas süsteem ehitada nii, et see oleks loogiline ning lihtsasti mõistetav (LEDide aadressid tuleb süstematiseerida ning luua elektroonika ja tarkvara vaheline loogika).

Lõputöö eesmärgiks on luua taevakaardil baseeruv valguspaneel, mis oma õdusa valguslahendusega on interjööri element. Eelkõige on programmeeritav taevakaart suurepärane õppevahend tähistava tähtkujude õppimiseks. Lisaks võib antud lõputööd kasutada kui juhendmaterjalina järgmiste valguspaneelide valmistamisel. Sellest tulenevalt on vaja täiustada LEDide juhtimise skeemi nii, et oleks tagatud paindlik

valgusejuhtimine programmikoodi muutmise kaudu. Lisaks peab ta olema soodsam ja lihtsam koostada, kui DMX kontrolleri ja Intelligentse LED kontrolleri puhul ning mis tagab Oma mõõtmetelt on valguspaneel loodud maksimaalselt suur, et sellele on võimalik paigaldada hõredalt suur kogus erinevaid tähti ning tähtkujusid, et need oleksid kergesti üksteisest eristatavad. Samas on mõõtmed mõistlikud, et paneeli on võimalik teisaldada ühest asukohast teise.

Lõputöö ülesanded:

1. Uurida prototüübi valmistamiseks vajalike komponentide alast kirjandust
2. Võrrelda erinevaid arendusplatvorme
3. Sobivate taevakehade valimine ning luua vastav kaart
4. Valmistada taevakaardi valguspaneel
5. Töötada välja juhtpaneeli skeem
6. Ühendada valgustus elektriskeemi
7. Töötada välja programm

Antud lõputöö koosneb sissejuhatusest, kahest peatükist, kasutatud kirjanduse loetelust ning lisadest. Esimeses peatükis käsitletakse programmeeritava taevakaardi loomiseks kasutatavate komponentide tööpõhimõtteid ja teemakohast kirjandust. Teises peatükis kirjutatakse detailsemalt programmeeritava taevakaardi erinevate osade valmistamisest.

# **1 PROGRAMMEERITAVA TAEVAKAARDI LOOMISEKS VAJALIK TEOORIA ÜLEVAADE**

## **1.1. Trükkplaadil kasutatavad komponendid**

Kogu töö koostamisel lähtub autor valgusdioodide võrgustiku juhtimise skeemist (multiplexing), mis üldjuhul on kolmemõõtmelise võrgustiku kujul, kuid ruumi kokkuhoiu ning kiudoptika mugavamaks paigaldamiseks tuleb ümber teha tasapinnaliseks. Juhtpaneeli valmistamiseks konstrueeritakse arendusplatvormi Arduino baasil trükkplaat, mille väljunditesse ühendatakse valmiv LED-paneel. Viimasel olevaid valgusdioode juhitakse mikroprotsessori ATMEGA328P programmeerimise läbi, et luua valgusmustrid. Kuna tegemist on väga võimalusterohke mikroprotsessoriga, mille lähtekood on vabavaraline, on sellele võimalik hiljem lisada lisakomponente ja soovi korral ühildada kogu süsteem kasvõi teleri puldi või juhtmevaba ühenduse kaudu telefoniga. Nii saab lihtsalt ning kiirelt kogu süsteemi juhtida mõne nutiseadme ekraanilt. Valmival trükiplaadil on kasutusel takistid, diodid, valgusdioidid, mikrokontrollerid ja nende pesad, kaablid ning transistorid.

### **1.1.1 Takisti**

Takisti on üks olulisemaid elektroonikakomponente ning seda kasutatakse elektrivoolu vähendamiseks. Tema tähtsaim tunnussuurus on takistus. Takistus võib olla muudetav (nt reostaat) või püsiv. Takisteid kasutatakse elektriahelais vajaliku suurusega voolu või pinge saamiseks (nt eeltakisti, regulaartakisti), voolu stabiliseerimiseks (nt ballasttakisti, barreter), elektriseadmete koormamiseks (koormustakisti) jm. Takistit saab kasutada vastava füüsikalise suuruse tajurina, kui takistus muutub mingi kindla välisteguri toimel (fototakisti, tensotakisti, termotakisti jm). Takistus on suurus, mis iseloomustab keskkonna, keha, edastuskanali, ahela vms takistavat või pidurdavat toimet liikumisele või energia ülekandele. [\[1\]\[2\]](#)

Antud lõputöös on takisti eesmärk kaitsta mikroprotsessoreid ja teisi komponente üleliigse pinge ja voolutugevuse eest.

### **1.1.2 Diod**

Diod on elektroonika element, mis laseb voolu läbi ainult ühte pidi. Seega on tegemist pooljuhiga. Dioode kasutades tuleb jälgida nende polaarsust. Dioode saab kasutada vahelduvvoolu alalisvooluks muundamiseks. Programmeeritava taevakaardi valmistamisel kasutatakse valgusdioode. Valgusdiod on diod, mis eraldab voolu endast läbi lastes valgust. Valgusdiod on madala võimsusega ja seetõttu parim valik võimalikult väikese juhtpaneeli ehitamiseks. [\[3\]\[4\]](#)

### **1.1.3 Transistor**

Transistor on pooljuht, mida kasutatakse elektriliste signaalide võimendamiseks, juhtimiseks ja genereerimiseks. Transistoreid saab kõige üldisemalt jaotada kaheks: bipolaartransistorid ning väljatransistorid. Antud töö puhul on kasutatud bipolaartransistoreid. Bipolaartransistoreid juhitakse voolu abil, väljatransistoreid aga pinge abil. Transistoris on tavaliselt kolm elektrijuhet, mida nimetatakse emitteriks, kollektoriks ja baasiks ehk juhtelektroodiks. Bipolaartransistorid jagunevad üldiselt kaheks: NPN- ja PNP-transistorid. Tüüp PNP või NPN näitab, mis pidi peavad pinged transistorile minema. NPN transistor vajab emitterile miinust ja kollektorile plussi. Transistori sulgemiseks peab baasile andma kas sama või negatiivsema pinge mis emitterile. Avamiseks tuleb anda baasile positiivsema pinge, mis emitterile. PNP transistori puhul on kõik vastupidine. Pinget ei tohi transistori peale otse lasta, sest siis põleb transistor läbi, vaid läbi eeltakisti. See tähendabki juhtimist vooluga. Mida suurem on vool baasil, seda rohkem transistor avaneb ning suureneb koormuse vool. Väike baasvoolu muutus põhjustab suurt kollektorvoolu muutust. Seetõttu sobib transistor võimendina kasutamiseks. [5][6] LEDide juhtpaneeli valmistamisel kasutatakse NPN transistoreid, et võimendada iga valgusallika grupi signaali.

### **1.1.4 ATMEGA328P-PU**

ATMEGA328P-PU on 8-bitine mikrokontroller, mille sisemälu maht on 32 kilobaiti. Antud mikrokontroller on vastupidav kergetele pingele ja voolu kõikumistele ning samas väga väikese voolutarbimisega (1MHz kiirusel, 1.8V pinge ning 25°C juures töötades 0,2mA). [7] Tegemist on kogu süsteemi peaprotsessoriga.

### **1.1.5 TLC5940/TLC5940NT**

TLC5940/TLC5940NT on 16 kanaliga mikrokontroller, mis kuulub ATMEGA328-le ning suudab juhtida LEDide sisse- ning väljalülitamist ja dimmerdumist (ereduse muutumist). Sisuliselt suurendab antud mikrokontroller ATMEGA328 sisendite või väljundite arvu ning on ühendatav jadamisi luues adresseeritava 16 sisendiga/väljundiga gruppidest koosneva ahela. Selle mikrokontrolleri abil on võimalik juhtida iga LEDi eraldi ning anda kindel kuni 120mA tugevusega vooluhulk 3-5,5V pinge juures. [8]

## **1.2. Arendusplatvormide võrdlus**

Arendusplatvorme on loodud väga palju, näiteks Arduino, Raspberry Pi, Sparkfun, micro:bit, Teensy jpt. Enamus arendusplatvormidel on vabavaraline tarkvara ning riistvara, mis teeb nad lihtsasti programmeeritavaks. Võrdlusesse on võetud kõige populaarsemad Arduino UNO ja Raspberry Pi, mis on oma mõõtudel ja võimekuselt väga sarnased ning lisaks Arduino Nano, mis on mõõtudel väiksem. Kõiki kolme on

kasutatakse erinevate digitaalsete seadmete loomiseks ning nendel võimalik programmeerida C-keeles.

### **1.2.1 Arduino UNO ja Arduino Nano**

Arduino UNO ja Arduino Nano on soodsaimad ning lihtsaimad arendusplatvormid, mis on mõeldud algajale elektroonikule. Arduino UNO saab Eestis soetada 26€ eest ([Oomipood](#)) ja Arduino Nano 16,29€ eest ([Farnell](#)). Arduino Nano suurimaks eeliseks on tema väga väikesed mõõtmed. Nende programmeerimine lihtsaim, sest programmeerimiskeelena on neile kohandatud C-keel, mis on samuti lihtsustatud, et õpilastel oleks lihtsam elektroonika ja programmeerimise printsiipe õppida. Arduino on enamlevinud arendusplatvorm ning tänu sellele on saadaval väga palju juhendeid erinevate seadmete valmistamiseks. Samuti on sellel platvormil suur kogus austajaid ning selle platvormi kasutajatel on võimalik leida töö käigus tekkivatele probleemidele kiiremini ja kergema vaevaga lahendused. Töö hõlbustamiseks on loodud palju blogisid ja foorumeid. Arduino koduleht on üles ehitatud nii, et algajal on lihtsam õppida. Samuti on loodud erinevaid komplekte õppuritele, kus on olemas kõik vajalik elektroonika algtõdede õppimiseks. Arduino võimaldab kasutada väga erinevat riistvara. Arduino UNO ja Arduino Nano on madalama võimekusega arendusplatvormid ning suutelised jooksutama vaid üht programmi korraga. [\[9\]](#)[\[10\]](#)

### **1.2.2 Teensy**

Teensy on välja toodud arendusplatvormidest vähem tuntud ja kallimate seas. Teensy on väike arendusplatvorm, millel on võimalik luua programme kasutades C-keelt. Teensy't on võimalik soetada Eestist 49€ eest ([Oomipood](#)). Teensy ühildub lihtsasti Arduino baasil kirjutatud programmidega. [\[11\]](#)

### **1.2.3 Raspberry PI**

Raspberry PI on uuem ning funktsionaalsem arendusplatvorm, kuid keerulisem programmeerida. Selle protsessor jooksutab Linuxi nimelist operatsioonisüsteemi, mis teeb antud seadmest väikese mõõtmelise arvutiga võrreldava seadme. Raspberry PI on Eestis soetatav 66,94€ ([Oomipood](#)) eest ning juurde tuleb osta SD-mälukaart, mille pealt kogu programmistik joosta saab. Raspberry PI on suurema töösagedusega ning suurema vahemäluga kui Arduino UNO. See seade tarbib rohkem voolu kui Arduino UNO. [\[12\]](#)

Antud lõputöö jaoks piisab lihtsast arendusplatvormist ning on oluline, et seade oleks pisike, kuna on vaja, et juhtpaneel on väikeste mõõtmetega. Seetõttu valis autor antud lõputöö jaoks Arduino Nano arendusplatvormi.

### 1.3. Arduino

Robotikas tuleb tihtipeale ette disaini- ja elektroonika ülesandeid, mille käigus soovitakse valmistada üsna keerukaid ülesandeid täitev seadeldis. Varasemalt oli vaja kogu riist- ja tarkvara igale prototüübile ise luua. Aastal 2005 loodi paari päevase töö käigus Itaalias Ivrea Interaktsioonidisaini instituudis õppejõudude Massimo Banzi ja David Cuartielles´i poolt lihtne ning võimalusterohke arendusplatvorm, et pakkuda oma kooli disainitudengitele lihtsat, odavat ja paindlikku elektroonika prototüüpimise seadeldist. Arduino saavutas väga kiiresti populaarsuse tudengite seas ning juba mõned kuud hiljem võttis meeskond vastu sadu tellimusi üle maailma. Arduinot on võimalik üpris lihtsalt ka ise valmistada. Arduino (vt [arduino.cc](http://arduino.cc)) on loodud eelkõige kunstnikke, disainereid ja hobielektronikuid silmas pidades, kuid on piisavalt võimas, et panna silmad särama ka nõudlikumal prototüübi loojal. Arduino on ühtlasi heaks alguspunktiks kõigile, kes soovivad tegeleda interaktiivsete objektidega alates robotitest, telekamängudest ja trummimasinatest kuni juhtmevabade kliimasensorite, 3D printerite ning nutikate valveseadmeteni välja.

Arduino puhul on tegemist kolme erineva tööriistaga:

1. Atmega328 mikrokontrolleril põhinev miniarvuti, mida võib osta ja ka ise ehitada erinevates suurustes vastavalt ideele ja eesmärkidele. Arduino sisenditeks võivad olla kõiksugused sensorid ja moodulid: puutekraanid, valgussensorid, liikumisandurid, potentsiomeetrid, GPS-moodulid, WIFI-moodulid jne. Samas väljunditeks võivad olla LED lambid, LCD-ekraanid, servomootorid, mootorid, kõlarid jne. [14]
2. Arduino programmeerimiskeel, mis põhineb C/C++ keelel ning kompileerija, mille abil luuakse mikrokontrollerile kood. [14]
3. Arduino, kui lihtne ja avatud lähtekoodiga arenduskeskkond, mis töötab Windowsi, Linuxi ning Mac operatsioonisüsteemide peal." [14]

Rebane ja Savi (2011)[14] sõnul on Arduino Creative Commons litsentsi all täiesti vabavaraline, avatud lähtekoodiga arendusplatvorm ning kõik skeemid on avalikud. Igaüks võib nende valmisskeemide põhjal lasta Hiinas valmistada tuhandeid arendusplatvorme ning neid müüa. Samuti on nad [14] öelnud, et Arduino on turul odavam arendusplatvorm, olles samal ajal avatud lähtekoodiga ning piisavalt võimekas, et ka keerukamaid projekte tema abil teostada. Arduino keel on niivõrd lihtne ja hästi dokumenteeritud, et sobib suurepäraselt algajatele sissejuhatuseks programmeerimisse ning piisavalt paindlik ja võimalusterohke, et edasijõudnutel oleks oma suuremaid projekte lihtne arendada. Keele õppimisel aitab veel kaasa suur ja abivalmis kogukond. Arduinosid on väga palju erinevaid ning vastavalt projektile tasub võrrelda erinevaid

mudeleid. Näiteks kõige pisemate projektide jaoks mõeldud Arduino Mini on võrreldav 2-eurose mündiga ning sobib projektidesse, kus kaalu ja ruumi kokkuhoid on väga olulised. Teine Arduino mudel on masinpestav LilyPad nime kandev arendusplatvorm, mida saab kasutada tekstiilikunsti projektide huvitavamaks muutmisel. Arduino Mega on suurte projektide jaoks sobiv platvorm, millele saab lisada erinevaid moodulplaate sensoritega. Samas kõige levinum on Arduino UNO, mis on oma suuruse ja võimekuse poolest kõige universaalsem ning pädlikum. Arduino UNOle saab lisada nii sensoreid kui ka lisamoduleid (ingl. shields). Lisamoodulid võivad anda juhtmevaba interneti, raadio, bluetooth ja mobiilsideühenduse ning erinevaid muid lisavõimalusi Arduino UNO kasutamiseks. Samuti väidavad Rebane ja Savi (2011)[14], et Arduino õppimine on muutunud aina populaarsemaks nii kesk- kui ka põhikoolis, mis arendab noorte silmaringi ja liidab endas nii praktilise tegevuse kui ka teoreetiliste programmeerimisteadmiste omandamise. Arduino kodulehel on selle tarbeks olemas õppematerjalide komplekt.

Käesolevas lõputöös on kasutatud Arduino Nano arendusplatvormi, sest sellega manipuleerimine on lihtne ja vajalik algkood, mida modifitseerida, on leitav internetist ning foorumitest. Arduino Nano on suhteliselt väike ning arendusplatvormil puuduvad üleliigsed komponendid. See tagab Arduino töökindluse ning vähendab võimalust konfliktide tekkeks programmi ning riistvara vahel.

## 1.4. Arduino keel

Arduino programmeerimiskeel põhineb C-keelel. C ja C++ on väga mitmekülgsed ja rohkete võimalustega ning annavad väga tugeva põhja programmeerimise õppimisel. Arduino keeles ei ole oluline teksti joondus. Erinevad programmi koodi plokid eraldatakse loogiliste sulgudega ({}), kuid üherealise ploki puhul ei ole sulud vajalikud. Kõik laused lõpetatakse alati semikooloniga (;). Arduino keeles on nimed tõstutundlikud (case sensitive). Kommenteeritud teksti on võimalik märgistada kahel viisil: esiteks, üherealine kommenteeritud teksti rida/osa ette kirjutatakse kaks kaldkriipsu (//) ning teiseks, pikema teksti kommenteerimiseks kasutatakse kommentaari lõiku. Kommentaari lõik algab kaldkriipsu ja tärniga (/\*) ning lõppeb täрни ja kaldkriipsuga (\*). Kommenteere kasutatakse oma koodi selgitamiseks ning see muudab koodi lihtsamalt loetavamaks ja mõistetavaks. [15]

Arduino keeles on kasutusel väga palju erinevaid funktsioone ja muutujaid, mille kasutamise viisid on täpsemalt välja toodud Arduino kodulehel. Vaata lähemalt juures olevalt lingilt: <https://www.arduino.cc/reference/en/>

## 1.5. Taevakaardi mudelid

Interneti avarustes surfates leiab palju erinevaid taevakaardi valguspaneele. Mõned neist on tehtud vaid LEDe kasutades, see aga tähendab, et iga täht tarbib eraldi voolu ning on kulukas ja vähem töökindel. Detailide rohkus on tihtipeale elektroonikas probleemide allikaks. Mida rohkem erinevaid osasid seadeldise valmistamisel on kasutatud, seda kiiremini ja suurema tõenäosusega võib miskit üles öelda ning vea otsimine võib olla vaearikas. Samas annavad detailsed seadmed rohkem võimalusi seadmega manipuleerida ning isegi ümber programmeerida. Enamus tähistaeva valguspaneelidel on siiski piiratult tööle pandud üksikute valgusallikatega, millede külge on kinnitatud kiudoptika kimbud, et voolutarbimist madalal hoida ning samas luua tähesära efekt.[\[16\]](#)[\[17\]](#)

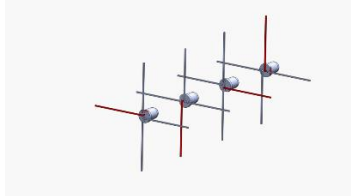
Eestis on ka sarnaseid töid tehtud, näiteks Fashion Indecor OÜ paigaldab täistaeva efektiga valguslahendusi lagedele [\[18\]](#). Suuremate kiibistikega tähistaevamudelite puhul on probleemiks jahutus. Tihti paigaldatakse tähistaeva mudeleid lagedesse või pildiraami, kus puudub piisav õhuringlus, et toiteplokki ning juhtkiipi jahutada. Samuti on probleeme olnud elektroonikapoole peitmisega. Seadmele peab säiluma ligipääsetavus, et vajadusel teda remontida või ümber häälestada. Enamjaolt soovitakse, et kogu süsteem oleks õhuke ning kergesti varjatav.

Lähim töö, mis antud teemaga haakub on LED cube - LED-kuubik, milles on kasutatud palju valgusallikaid, ning tagatud nende programmeerimise võimalus ja suhteliselt vähene detailide arv. Samas on ka neid valmistatud erinevalt.[\[19\]](#)[\[20\]](#) Osadel on transistorite (LEDide jaoks antava signaali võimendamiseks) kogus üüratult suur, sest võimendatakse iga LED tulba värvidele minevat signaali, teiste puhul on lahendatud probleem nii, et võimendatakse vaid LEDide korrustele minevat signaali.

Näiteks, kui võtta 8x8x8 LED kuubik saab näha 512 erinevat valgusallikat, millel kõigil on anood- ja katoodeühendused. Samas multiplexi meetodil saab taandada anoodide arvu 8 peale ning katoodeide arvu 64 peale. Ehk siis 72 ühendust eelneva 1024 asemel. Samas, kui võtta aga RGB LEDid, milledest igaühel on 4 jalga (1 anood ja 3 katodi) selgub, et kokku tuleks teha 2048 ühendust. Multipleksides pääseb 192 katoodeühenduse ja 8 anoodühendusega. Kokku 200 ühendust. Ühenduste arv on siiski liiga suur, et kõik ühe mikroprotsessori külge ühendada. Selle probleemi lahendamiseks on kasutatud Shift register TLC5940 nime kandvaid mikroprotsessoreid, mis omavad digitaalse informatsiooni sisendit ning väljundit. Neid ahelasse ühendades ning mikroprotsessorile salvestatavas programmis neile viidates on võimalik selgeks teha, kui palju on ahelas eraldi adresseeritavaid sisendi või väljundi ühendusi. Jääb vaid üle tagada piisav voolutugevus ning pinge, et valgusallikad ühtlaselt põleksid.



Töö käigus on autor leidnud ka teise meetodi LEDide grupiti juhtimiseks: charlieplexing. Antud tehnika lubab juhtida LEDe ilma TLC5940 mikroprotsessorkive kasutamata. Siiski on antud meetodika selle töö puhul probleemne. Nimelt ei ole võimalik LEDe ühendada tasapinnale gruppidesse, et nende külge kiudoptikat kinnitada. Kui multiplexing puhul olid LEDide jalad ühendatud paralleelselt (plussid omavahel kokku ja iga värvitooni jalad värvi järgi kokku), siis charlieplexing puhul luuakse LEDi tornide süsteem pöörates LEDe 90° nii, et tornis olevate LEDide jalad on segamini ühenduses (vt Pilt1.5-1). Punased jalad tähistavad pluss klemmi.[\[21\]](#)



Pilt 1.5-1 Ühe LEDi tulba Charlieplexing

## **2 PROGRAMMEERITAVA TAEVAKAARDI KONSTRUEERIMINE**

### **2.1. Juhtpaneel**

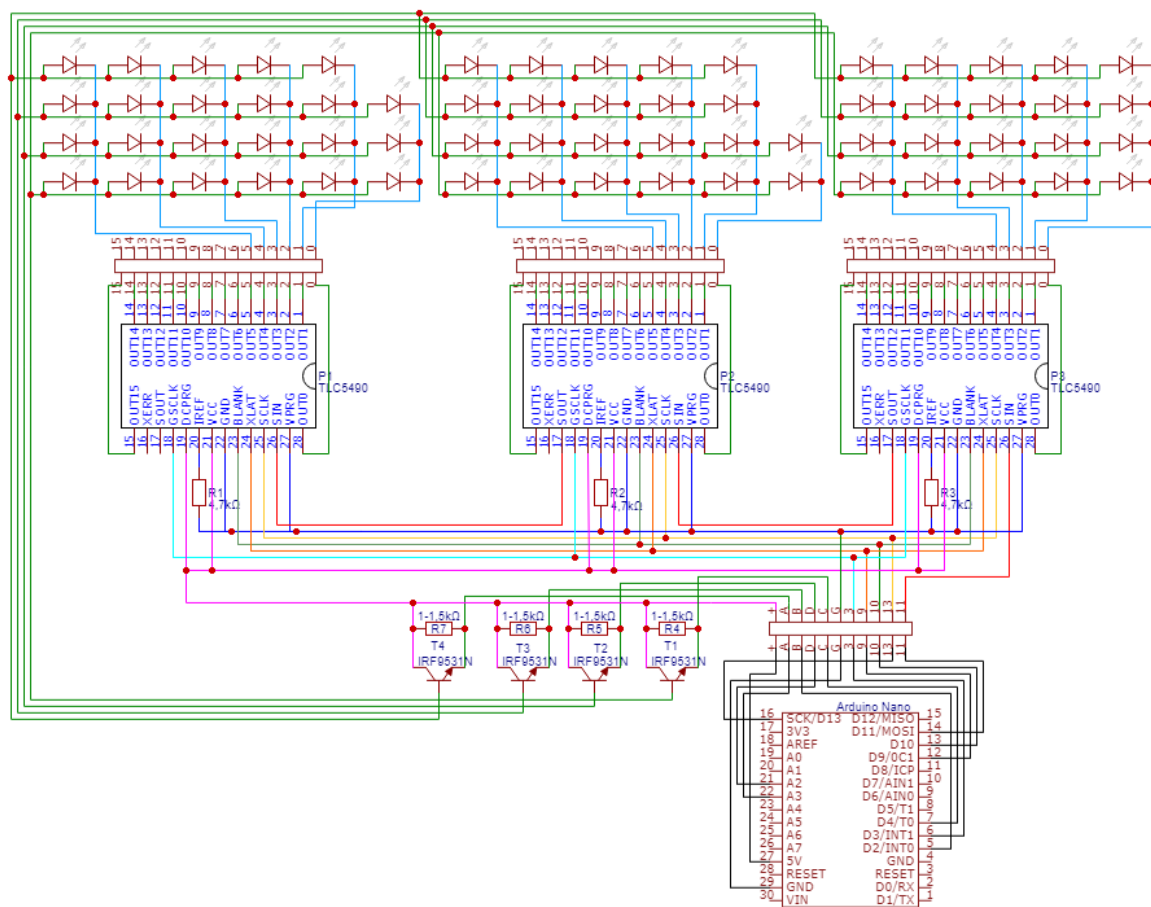
LED valgust on võimalik juhtida väga mitmel erineval viisil. Tänapäeva ehituses kasutatakse peamiselt LED ribasid ning pirne, mille juhtimiseks kasutatakse kas DALI juhtimissüsteeme, kus on digitaalsed signaali vastuvõtjad ning saatjad või DMX juhtimist, kus määratakse igale valgustile aadress ning juhitakse signaali läbi valguspuldi. Valgustites võib olla mitmeid LEDide grupe, kuid valgusallikad (LEDid) ei ole ükshaaval juhitavad. Erandiks on Intelligentsed LED ribad, kus igale LEDile on lisatud pisike adresseeritav kivi. Antud töös aga ei saa kasutada intelligentseid LED ribasid, sest nendel olevad LEDid on lapikud ning kiudoptika kiudude punktide kinnitamine nendele on keerukas. Samuti on raske remontida riba kui mõni LED peaks ajapikku läbi põlema, sest kiudoptika kimbud tuleks LEDide külge liimida. Seega tuleb valmistada 5 mm läbimõõduga LEDidest maatriks, ning iga eraldi juhitava LEDi külge kinnitada termokahaneva rüüsiga kiudoptika kiudude kimp. See tagab lihtsa remontimise võimaluse. Nimelt saab vajadusel termokahaneva rüüsi lahti lõigata ning LEDi välja vahetada.

Juhtpaneel on programmeeritava taevakaardi valguspaneeli üks olulisim osa. See koosneb kahest osast: Arduinol baseeruvast juhtpaneeli plaadist ning sellega ühendatud LEDide plaadist. Juhtpaneel ehk kontrollor on juhitav Arduino kaudu ATMEGA328P-PU protsessorkivile laetud programmidega. Programmid kirjutatakse Arduinole kohandatud C-keeles ning on üpris lihtsa struktuuriga. Samas on kogu süsteemi loomise juures

tõenäoliselt kõige keerulisem osa programmi loomine ning selle komponentide (käskluste) mõistmine.

Juhtpaneeli lihtsustamise käigus on autor pööranud tähelepanu LED-kuubiku tasapinnalise maatriksi konstrueerimisele ning LEDide grupeerimisele ja kolme TLC5940 Shift registri vahele jaotamisele. LED-kuubiku juhtpaneelist erineb antud trükkplaat veel selle poolest, et eemaldatud on üleliigsed kondenssaatorid ning 2 kOhm takistite asemele on ühendatud 4,7 kOhm takistid, et säilitada võimalikult suur valgustugevus ning kivide eluiga. Arduino on integreeritud juhtpaneeli, seega on võimalik tõsta eelprogrammeeritud animatsioonidega kivi lihtsalt trükkplaadi sisse ning kogu süsteem töötab. Samuti on säilitatud võimalus programmeerida kivi ümber ilma seda plaadilt eemaldamata. Samuti saab kivi programmeerida ühendades ilma kivita Arduino paralleelselt kiviga juhtpaneelil.

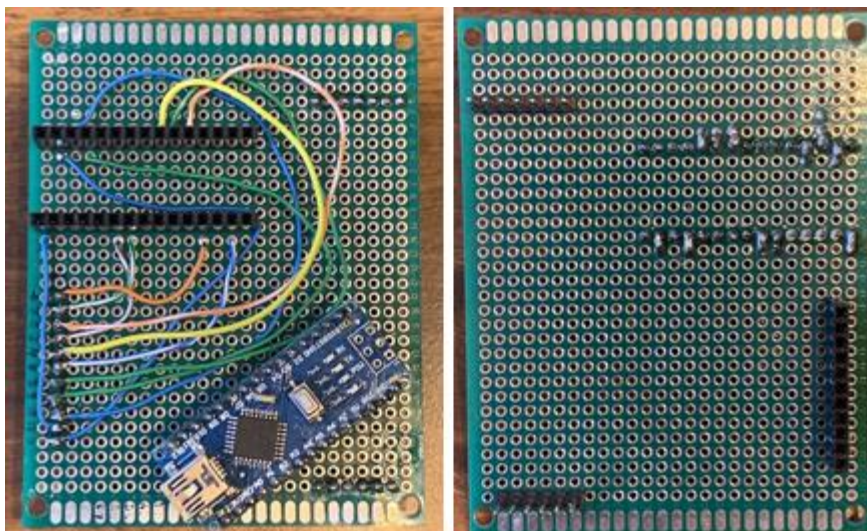
Programmeeritava taevakaardi erinevate paneelide valmistamiseks vajalikud komponendid ning nende maksumus on välja toodud lisades (vt [Lisa 1](#)). Juhtpaneeli valmistamise skeem (vt Pilt 2.1-1) on koostatud kasutades programmi EasyEDA.



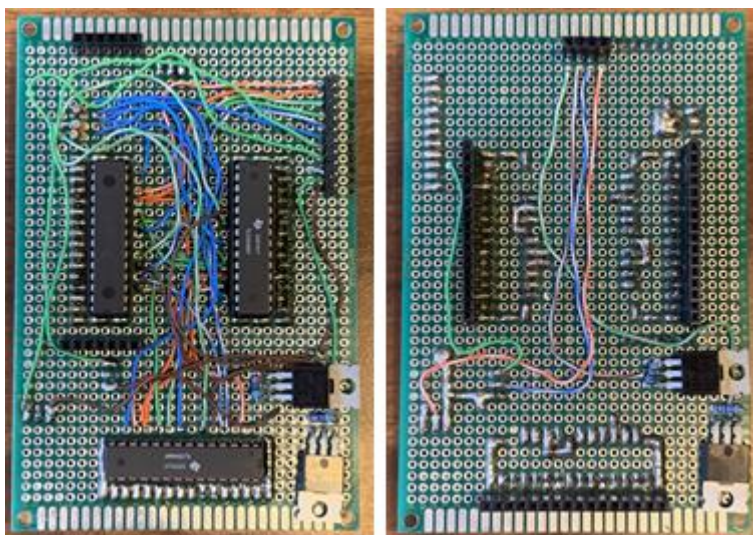
Pilt 2.1-1 Juht- ja LEDpaneeli valmistamise skeem

LED juhtpaneeli valmistamisel võib mikroprotsessorite väljundite/sisendite ühendamiseks kasutada nii juhtmeid, kui ka jooteradasid. Jooteradade tegemisel tuleb parima tulemuse saavutamiseks ühendada esmalt trükkplaadil olevad vasetatud ruudud 2-kaupa ning hiljem joota nad ühtseks rajaks kokku. Nii ei teki ülisuuri tinakogumeid ning rada jääb ühtlase laiusega.

Juhtpaneeli pildid eest ja tagant (vt Pilt 2.1-2 ja Pilt 2.1-3). Esimesele on ühendatud Arduino Nano ja teisele TLC5940NT kivid, transistorid ning takistid.

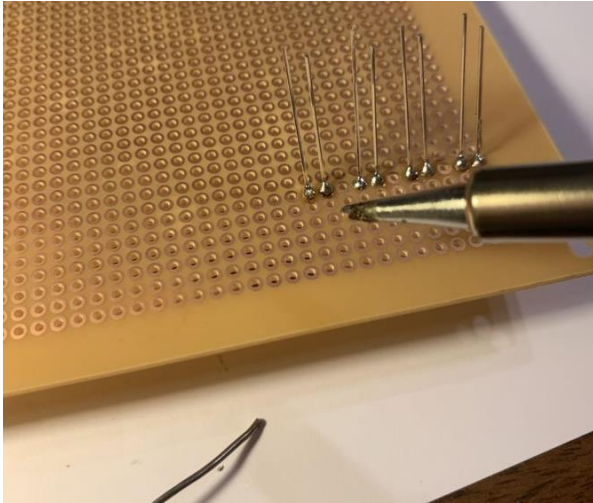


Pilt 2.1-2 Juhtpaneeli esimene osa pealt ja alt



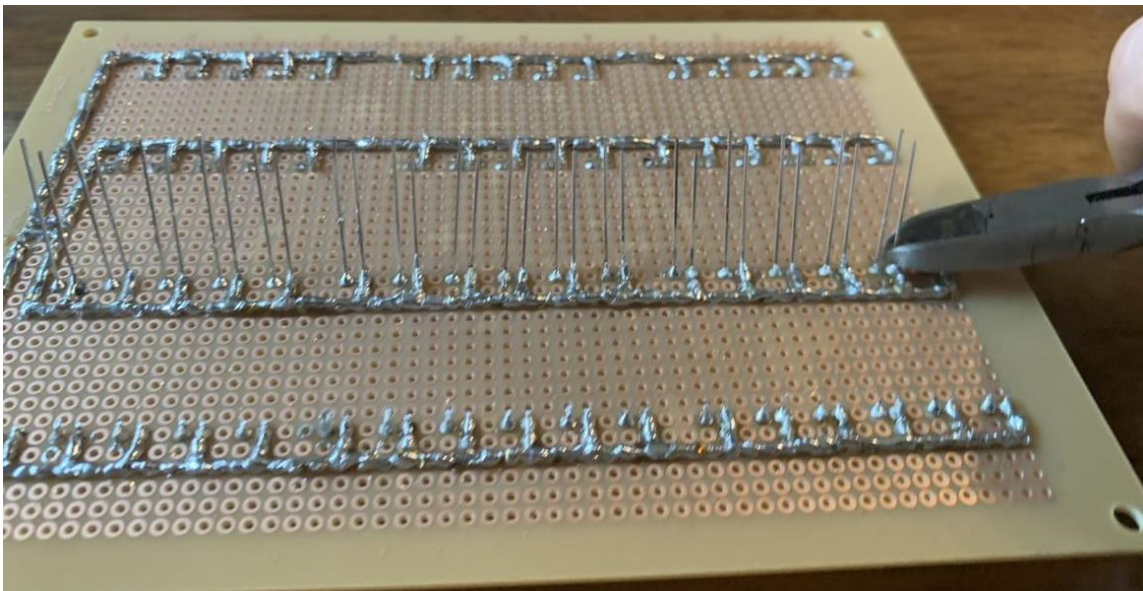
Pilt 2.1-3 Juhtpaneeli teine osa pealt ja alt

LEDide paneelile jootmisel tuleb arvestada, et LED kardab kõrget temperatuuri. Seetõttu ei tohiks LEDi jalgu jootekolviga kuumutada mitte rohkem kui 1-2 sekundit. Iga jootmise vahel tuleks liitekohal lasta jahtuda või jahtumise kiirendamiseks puhuda ühenduskohale (vt Pilt 2.1-4).



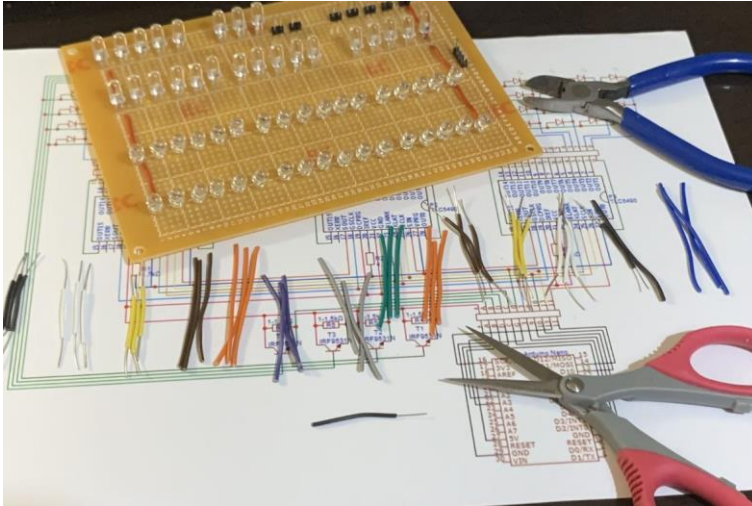
Pilt 2.1-4 LEDide jootmine

Paneelide mugavamaks kasutamiseks on vaja kõik üleliigsed juhtmed ja LEDide jalad ära lõigata lõiketangidega (vt Pilt 2.1-5).

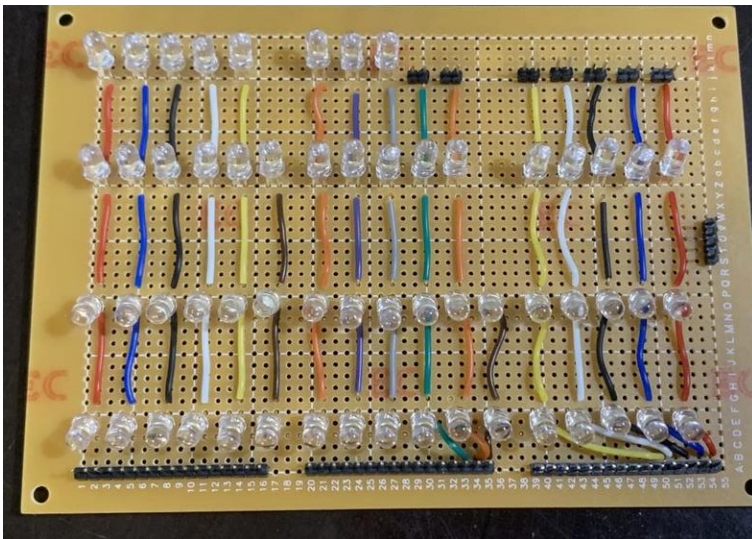


Pilt 2.1-5 LEDide jalgade lõikamine

LEDide miinus jalad on ühendatud kokku tinutatud jooteradadena ning pluss jalad juhtme juppidega. Vaata lähemalt allolevatelt piltidelt (vt Pilt 2.1-6 ja Pilt 2.1-7).



Pilt 2.1-6 LEDpaneeli juhtmete ettevalmistamine



Pilt 2.1-7 LED paneel pealt

LEDide ja kiudoptika ühendamisel tuleb kiudoptika kimbud eelnevalt õhema termokahaneva rüüsiga kokku fikseerida ning toestamiseks kasutada kuuma liimi, kimbust olevate kiudude vahel, et keskmised kiud ei tuleks ajapikku termokahaneva rüüsi välja. Suuremate kimpude (enam kui 14 kiudu) puhul tuleb arvestada, et valgus on intensiivsem LEDi tsentrisse jäävatel kiududel. LEDi tsentrist eemal olevad kiud annavad ekraanil tuhmimat valgust. Samuti tuleb tagada termokahaneva rüüsi kuumutamisel, et see kokku tõmbudes ei jätaks LEDi ja kiudude vahele lõtku, sest see võib tekitada ekraanil ebaühtlase valguse või takistada valguse kiududesse jõudmist.

Kiudoptika kiude valides (välja mõõtes) tuleb arvestada, et kiud olgu pigem pikem (30-50 cm), kui lühem. Nii tagatakse mugavam kiudude ekraanile ühendamine ning kiud ei jää pingele alla. Kui kogu taevakaardi paneel on LED paneeliga ühendatud, lõigatakse väljaulatuvad kiud ekraani pinnaga tasaseks. Et ekraani oleks meeldiv käega

puudutada, sulatatakse jootekolviga kiudude otsad lõplikult ekraani pinnaga tasaseks. Sulatamisel laieneb kiu ots piisavalt, et täita ekraani pinnal oleva ava serv ning selle pind jääb ühtlane. Vaata lähemalt kiudoptika paigaldamist allolevalt pildilt (vt Pilt 2.1-8).



Pilt 2.1-8 Kiudoptika kiu paigaldamine valguspaneeli

Antud töö puhul on lähtunud juhtsüsteemi konstrueerides LED kuubiku tööpõhimõttest. Kogu elektroonika sarnaneb LED kuubiku juhtsüsteemile. Samuti on programmeerimine sarnane LED kuubiku programmeerimisega. Põhiline erinevus seisneb selles, et LED kuubiku puhul püütakse valguse ning ruumiliste visuaalsete efektidega luua keerukas geomeetriline kunstiteos, kus iga muutust võetakse kui kaadrit filmilõiguse. Programmeeritava taevakaardi valguspaneeli puhul ei püüta luua ruumilist, vaid tasapinnalist mustrite vahetumist. Sellisel juhul ei ole tarvis ruumiliselt LEDe paigutada. Samuti raskendab kuubiku kujul olev LED paneel kiudoptika kimpude kinnitamist ning seadme remonditavust. Kuubiku puhul ei ole võimalik ilma endoskoopiliste abivahenditeta selle keskosas olevaid LEDe vahetada. Samuti on keerukas kiudoptikat paigal hoidvat termokahanevat rüüsi kahanema panna ilma jooteliiteid lahti sulatamata. Seetõttu on autor viinud kuubiku süsteemi tasapinnaliseks, kuid seejuures säilitanud süsteemis olevate valgusallikate järjekorra ning juhitavuse ja adresseeritavuse. See on ka antud töö suurim lihtsustus, mis loob võimaluse taevakaardi valguspaneeli valmistamiseks.

Taevakaardi valguspaneeli valmistamisel on suur roll programmeerimisel. Antud programmi tuleb iga valmiva seadme (olgu see siis taevakaardi valguspaneel või LED kuubik) jaoks kalibreerida (LEDide aadressid täpsustada). Sellisel juhul on võimalik luua loogilisi ning efekteid visuaalseid programme.

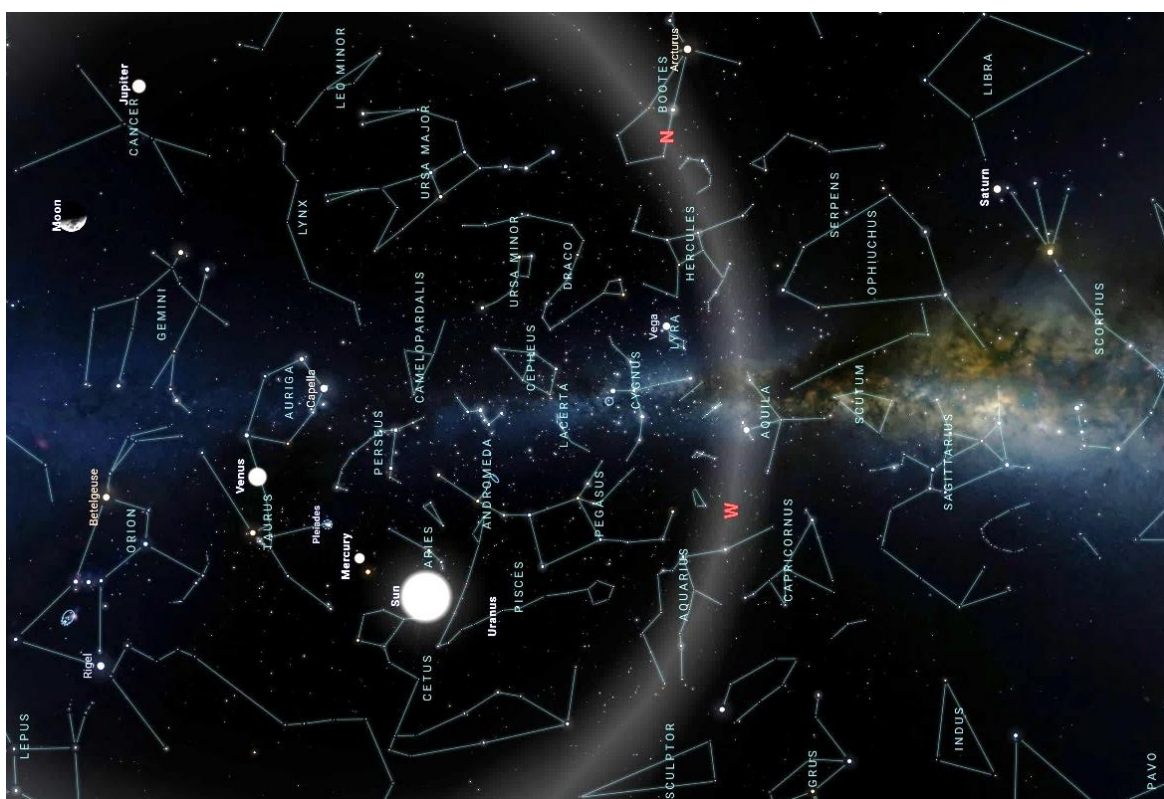
Programmi põhiosas võib märkide /\* ja \*/ abil elimineerida need programmilõigud, mida ei soovita näha ning numbreid muutes kogu valgusmustrit muuta. Kokku on autor kirjutanud mitu erinevat programmilõiku, mida saab lõpmatuseni kordama panna,

nende järjekorda muuta ning neid modifitseerida või nende järgi hoopis uue programmi kirjutada.

Programmi kirjutamiseks on tarvis LEDid aadresseerida. Kuna TLC5940NT kivi on 16 kanaliga ning antud töös on LEDe (65tk) kokku vähem, siis jääb TLC puhul mõned kanalid kasutamata. See jätab võimaluse hiljem lisada eraldi mõnele erilisele tähele oma LED ja optiline kaabel. Nt selliselt toimides on võimalik eraldi sisse/välja lülitada põhjanaela. Igale TLC5940NT-le väljundile määratakse tähis A-st kuni P-ni. TLC6590NT kivi väljundeid loetakse 0-st kuni 15-ni.

## 2.2. Taevakaardi valguspaneel

Taevakaardi valguspaneeli valmistamisel on kasutatud veebis olevat taevakaardi generaatorit Stellarium Web [\[22\]](#), kus on valitud kindel kuupäev ning kaarti on keeratud vastavalt, et autori perekonnaliikmete tähtkujud oleksid nähtavad kaardil (vt Pilt 2.2-1).



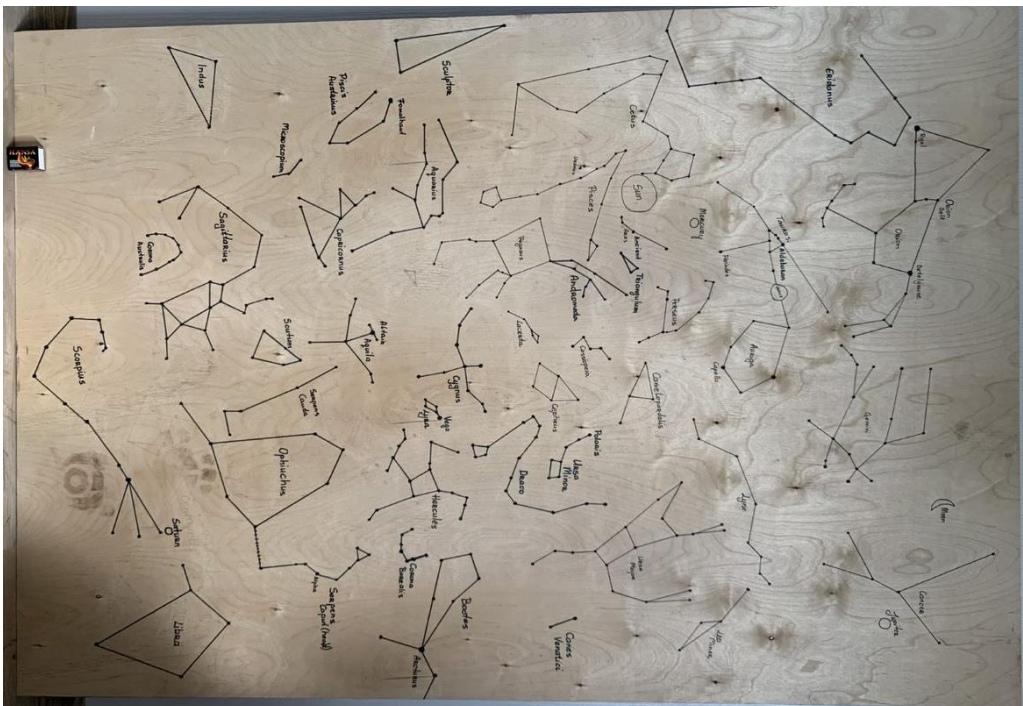
Pilt 2.2-1 Taevakaart Stellarium Web'ist

Taevakaardi valguspaneeli presenteerimiseks valmistatakse 10 mm kase vineerplaadist alus (mõõdud 1040 mm x 1525 mm). Valguspaneeli suurusest tulenevalt leidis autor, et kõige mõistlikum on tähed märkida paneelile hariliku pliatsiga projektori abiga (vt Pilt 2.2-2).



Pilt 2.2-2 Taevakaardi märkimine vineerpaneelile

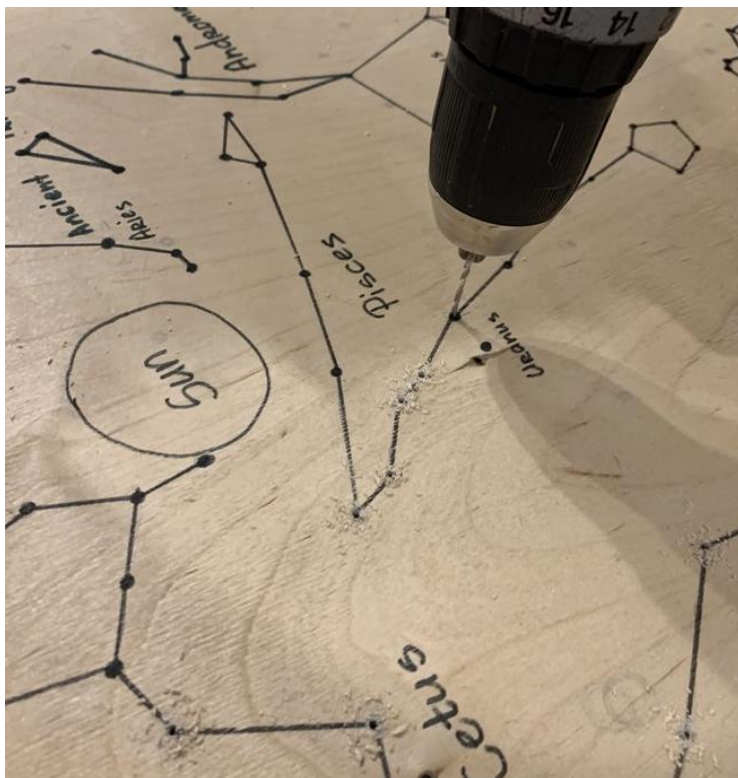
Tähed grupeeritakse tähtkujudesse ning joonistatakse musta viltpliatsiga joonlaua abil tähtede vahelised mõttelised jooned ja tähtkujude nimed, et tähtkujud selgelt eristatavad oleksid. Vaata lähemalt allolevalt pildilt (vt Pilt 2.2-3), kuhu on lisatud tikutoos, et mõista paremini paneeli mõõtmeid.



Pilt 2.2-3 Tähtkujude kontuuride joonistamine

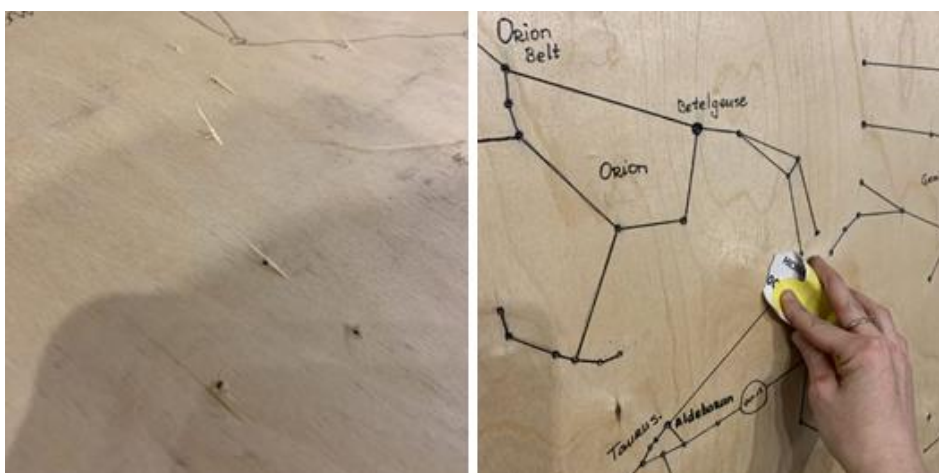


Paneeli sisse puuritakse kiudoptika kiudude läbimõõtu arvestades tähtede jaoks augud (vt Pilt 2.2-4). Tähtkujude tähtede jaoks on kasutatud 2mm puuri, suurematele tähtedele (nt põhjanael) 2,5mm puuri ning planeetide jaoks 5mm puuri. Planeetide LEDid ühendatakse juhtmetega LEDide plaadi küljest vineerpaneelil oleva LEDi külge (LEDid liimitakse liimipüstoli abiga paneelile).



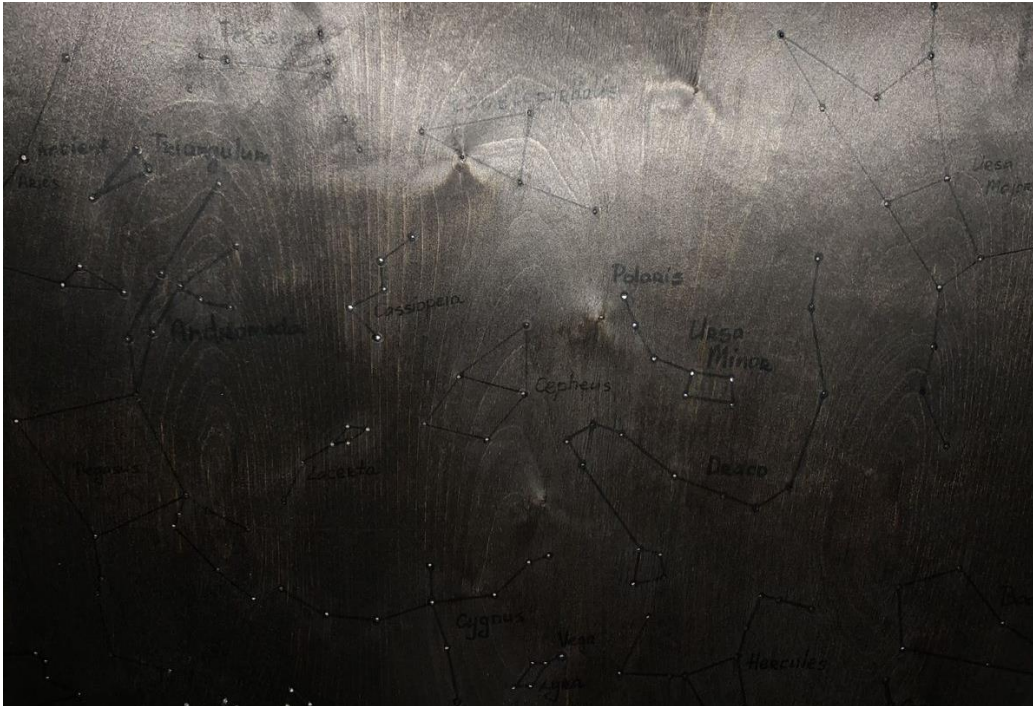
Pilt 2.2-4 Aukude puurimine tähtedele

Aukude puurimisel vineeriplaati ilmnes probleem. Puurimine lõhkus valguspaneeli tagumisel küljel vineeri pealmist kihti. Sellest tulenevalt oli vaja kõik puuritud augud liivapaberiga kergelt üle lihvida (vt Pilt 2.2-5).



Pilt 2.2-5 Puurimisest tekkinud defektide lihvimine

Efektsema pildi loomiseks värvitakse valguspaneel mustaks, et öisele taevale kohaselt tähed säraksid mustas taevas (vt Pilt 2.2-6). Paneeli värv on valitud selliselt, et valguspaneel mõjuks mustana ning vineerpinna puidu tekstuur ja varasemalt joonistatud tähtkujude jooned on jäävad nähtavale (vt Pilt 2.2-7). Hiljem joonistatakse need jooned ning tähtkujude nimed üle UV-aktiivse valge värviga. Selliselt toimides on tähtkujud nähtavad nii päevavalguses kui ka öhtul UV-lambiga ruumi valgustades.



Pilt 2.2-6 Valgustatud valguspaneel



Pilt 2.2-7 Värvitud valguspaneel. Valgustatud tähtedega valgus paneel

Taevakaardi mudeli animatsioonide tarbeks on tähtkujud LEDide külge optilise valguskaabliga ühendatud selliselt, et nad on järjestatud ekraani keskelt välja poole liikumise järgi. Selliselt on lihtsam üle taeva minevaid valguslaineid luua.

Tähtkujud ja tähed on järjestatud alustades keskelt ning liikudes välja poole. Tähtkujude juures sulgudes on märgitud tähtede arv tähtkujus ehk mitu optilise valguskaabli kiudu on vaja ühendada LEDi külge. Planeedid on suuremad kui tähed. Sellest tulenevalt liimitakse planeetide LEDid otse valguspaneeli külge ning ühendatakse juhtmetega LED paneelile. Tabelis on esitatud kõikide tähtkujude, tähtede ning planeetide nimekiri, mis on valguspaneelil kujutatud:

<b>Tähtkujud</b>		<b>Planeedid (LEDid)</b>
1.	Cassiopeia (5)	Saturn
2.	Camelopardalis (4)	Uranus
3.	Perseus (12)	Sun
4.	Trianglum (3)	Mercury
5.	Andromeda (9)	Moon
6.	Pegasus (14)	Jupiter
7.	Lancerta (6)	Venus
8.	Cepheus (5)	
9.	Polaris (Ursa Minor'i täht)	
10.	Ursa Minor (7)	
11.	Draco (15)	
12.	Cygnus (10)	
13.	Pisces (18)	
14.	Ancient (Aries'e täht)	
15.	Aries (4)	
16.	Pleiades (Taurus'e täht)	
17.	Aldebaran (Taurus'e täht)	
18.	Taurus (11)	
19.	Auriga (6)	
20.	Lynx (8)	
21.	Ursa Major (7)	
22.	Ursa Major complete (18)	
23.	Hercules (18)	
24.	Vega (Lyra täht)	
25.	Lyra (5)	
26.	Delphinus (5)	

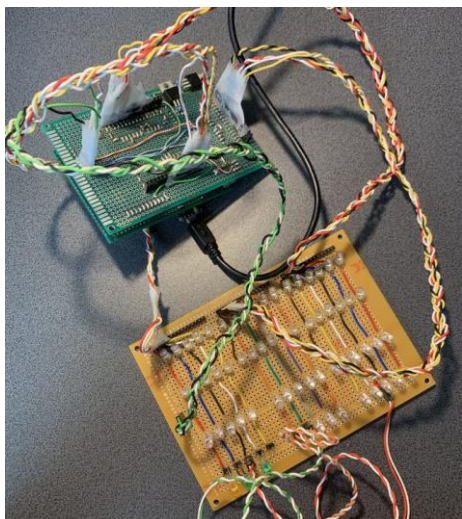
27.	Equuleus (4)	
28.	Altair (Aquila täht)	
29.	Aquila (9)	
30.	Aquarius (15)	
31.	Cetus (19)	
32.	Eridanus (12)	
33.	Orion Belt (3)	
34.	Rigel (Orion'i täht)	
35.	Betelgeuse (Orion'i täht)	
36.	Orion (20)	
37.	Gemeni (18)	
38.	Cancer (6)	
39.	Leo Minor (4)	
40.	Canes Venatici (2)	
41.	Arcturus (Bootes'e täht)	
42.	Bootes (8)	
43.	Corona Borealis (7)	
44.	Serpens Caput (9)	
45.	Ophiuchus (7)	
46.	Serpens Cauda (5)	
47.	Scutum (5)	
48.	Capricornus (10)	
49.	Fomalhaut (Piscis Australus'e täht)	
50.	Piscis Austrinus (7)	
51.	Sculptor (3)	
52.	Grus (6)	
53.	Microscopium (18)	
54.	Indus (3)	
55.	Sagittarius (21)	
56.	Corona Australis (10)	
57.	Scorpius (14)	
58.	Libra (5)	

### 2.3. Juhtpaneeli programmeerimine

Programmeerides on kõige olulisem mõista Arduino keelt ning analüüsida varasemaid sarnaseid programme. TLC5940 shift registrit kasutavaid programme on loodud mitmeid. Enamik neist on küll servomootorite juhtimiseks mõeldud, kuid on loodud ka

LEDide juhtimiseks mõningad programmid. Esmalt tuleb lisada Arduino kataloogi (library) TLC5940 kataloog, mille leiab Arduino kodulehelt ning avada TLC\_config.h nimeline fail Wordpad või Notepad++ nimelise programmi abil. Selles kataloogis tuleb muuta rida: "#define NUM\_TLCS 1" Antud reas tuleb number "1" asendada "3". Sellega määratakse ära kivide arv ahelas.

Programmikoodi testimiseks on juhtpaneel ja LED paneel ühendatud Jumper-juhtmetega (vt Pilt 2.3-1).



Pilt 2.3-1 Programmikoodi testimine

Programmeeritava taevakaardi Arduino programmikoodi võimalused:

Funktsioon	Koodi jupp
Kõikide tähtede, tähtkujude ja planeetide sisse lülitamine	<pre>//Turn ON all void AllON(){   digitalWrite(LAYER_ONE, HIGH);   digitalWrite(LAYER_TWO, HIGH);   digitalWrite(LAYER_THREE, HIGH);   digitalWrite(LAYER_FOUR, HIGH);   Tlc.clear();   for (int i = 0; i&lt;65; i++){     Tlc.set(i, 4095);   }   Tlc.update(); }</pre>
Lülita sisse ainult tähed ja tähtkujud	<pre>//Turn ON all constellations and stars void StarsON(){   digitalWrite(LAYER_ONE, HIGH);   digitalWrite(LAYER_TWO, HIGH);   digitalWrite(LAYER_THREE, HIGH);   digitalWrite(LAYER_FOUR, HIGH);   Tlc.clear();   for (int i = 0; i&lt;58; i++){     Tlc.set(i, 4095);   }   Tlc.update(); }</pre>

Lülita sisse/välja ainult planeedid	<pre>//Turn ON all planets void PlanetsON() {   SaturnON();   UranusON();   SunON();   MercuryON();   MoonON();   JupiterON();   VenusON(); }</pre>
Lülita sisse/välja tähtkujud järjest alustades keskelt ning liikudes väljapoole.	<pre>//Turn ON all One by One void AllOneByOneON() {   digitalWrite(LAYER_ONE, HIGH);   digitalWrite(LAYER_TWO, HIGH);   digitalWrite(LAYER_THREE, HIGH);   digitalWrite(LAYER_FOUR, HIGH);   Tlc.clear();   for (int i = 0; i&lt;58; i++){     Tlc.set(i, 4095);     delay(1000); //1sek=1000ms   }   for (int i = 59; i&lt;65; i++){     Tlc.set(i, 1000); //For planets less light     delay(1000); //1sek=1000ms   }   Tlc.update(); }</pre>

Programmikoodi tervikuna on võimalik lähemalt vaadata lisadest. (vt [Lisa2](#))

## 2.4. Juhtpaneeli valmistamise probleemid

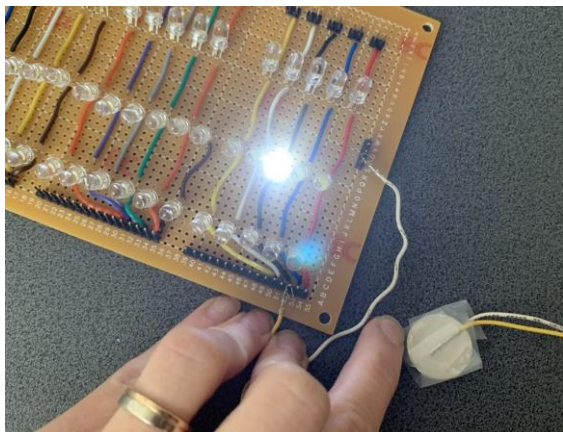
### Juhtpaneeli valmistamise käigus tekkis mitmeid probleeme:

1. Jooteradade kokku sulamine. Jootekolvi otsa suurus on jootmistöödel väga oluline. Laia otsaga on joota hea, sest temperatuur on kõrgem ning tina sulab kiiremini, samas on raske teha peenemaid jooteid. Jootmise sujuvamaks muutmiseks ostis autor uue jootekolbi, millel on väike ots ning temperatuuri on võimalik reguleerida.
2. Mittetöötavad LEDid. LED paneeli puhul selgus seadme katsetamisel, et osad LEDid ei töötanud. Nende välja vahetamine on aeganõudev ning tülikas. Mõistlik on LEDe testida enne jootmist. Hea võimalus on selleks ühendada LED vähemalt 3V patareiga (vt Pilt 2.4-1), nt CR2025 patarei.



Pilt 2.4-1 LEDide kontrollimine patareiga

3. Lühised. Kõik kõrvuti asetsevad ühendused tuleb enne esimest käivitust testeriga üle kontrollida, vastasel juhul võib mõni kivi esimese paari sekundi jooksul läbi põleda. Lihtsaim viis on ühendusi kontrollida CR2025 patarei abil (vt Pilt 2.4-2).



Pilt 2.4-2 Jooteradade ühenduste kontrollimine

4. Transistorite ühendamise Arduino ja LED paneeliga oli samuti probleemne. Nimelt lekkis nendest ATMEGA328-le voolu ning see kuumutas mikroprotsessorit kuni viimane oma töö lõpetas. Antud probleemi lahendamiseks ühendati transistorid järgnevalt: Esimese jala külge ühendatakse Arduinost tulev signaal, Teisest jalast läheb vool edasi LEDide kihile ning kolmas jalg ühendub +5V külge. Esimese ja viimase jala vahele ühendatakse 1-1,5 kOhm takisti, et kaitsta Arduinot läbi põlemise eest ning, et tagada seadme välja lülitumisel kohene valguse kustumine.
5. Valminud juhtpaneeli eesmärk on juhtida LEDe sisse ning välja lülitades tasapinnal asetseval taevakaardil olevate tähtkujude animatsioone. Selle jaoks tuleb LED paneel teha samuti tasapinnaline. Samas tuleb säilitada multiplex võrgustik, et LEDe saaks sarnaselt LED-kuubikule juhtida. Sellega analoogselt on valmistatud tasapinnaline LED paneel. Tasapinnalise LED paneeli küljes olevate LEDide külge kinnitatakse kiudoptika kiudude kimbud.

#### **Kiudoptika kimpude LEDidega ühendamisel tekkinud probleemid:**

1. Kiudoptika kimbud on liiga lühikesed. Kiudoptika kimpude pikkus peaks olema varuga, et tagada lõtk ning mugavam kimpude kinnitamine LEDide külge.
2. Kiudoptika kinnitamine valguspaneelile (vineerplaat, lagi, laua pind, valguskasti pind jne). Iga liim ei sobi sõltuvalt valguspaneeli pinna materjalist. Superliim sobib puidu ning metalli puhul, kuid näiteks pleksiklaasi puhul jätab see inetud valged plekid. Soovitatav on kiudude kinnitamiseks kasutada kuumaliimi. Kuumaliim vajab oskuslikku kasutamist. Niisama liimides kiudude kimpe paigal ei hoia ning liim tuleb pärast jahtumist pinna küljest lahti. Selle vältimiseks tuleb

kiudude ekraani materjali sisse jääv osa samuti liimiseks teha ehk mõned korrad kiudu avas liigutada nii, et see määriks liimi ka ava seintele laiali. Nii luuakse kindel ühendus.

3. Tänu eesmärgile luua miniatuurne skeem on antud töö puhul LEDid teineteisele väga lähedal. See teeb kiudude kimpude LEDide külge kinnitamise keerukamaks. Kogu kinnitusprotsessi lihtsustamiseks võib LEDide ühendamisel trükkplaadiga kasutada juhtmeid, mitte neid otse trükkplaadile joota nii, nagu antud töös seda tehtud on. LEDid tuleb nummerdada ning igale numbrile määrata tähtkuju. Järgmisena tuleb kiudude, mille pikkused on umbes 30-40 cm, mõlemad otsad ära puhastada nii, et esimene ots, mis kinnitatakse LEDi külge oleks 2-2,5 cm pikkuselt puhastatud ning teine ots kuni 5 cm pikkuselt puhastatud. Kimbud kinnitatakse esmalt LEDide külge ning hiljem ekraani külge. LEDide külge kinnitamisel kasutatakse liimikihiga termokahanevat rüüsi ning termokahaneva rüüsi kohalt fikseeritakse kiud kuumaliimiga, et ükski kiud termokahanevast rüüsisist välja ei tuleks.



## KOKKUVÕTE

Käesolevas töös keskendutakse LEDide juhtimisele tuginedes Arduino Nano nimelise arendusplatvormi rakendamisele programmeeritava taevakaardi valguspaneeli juhtsüsteemi loomisel. Selgitatakse vajaminevaid materjale ning tuuakse välja töö käigus ilmnedavad probleemid ning nende lahendused.

Töö esimeses peatükis käsitletakse olemas olevat teemakohast kirjandust ning tehnoloogia poolest üpris sarnase ehitusega LED kuubiku tööpõhimõtet. Samuti tuuakse välja erinevate arendusplatvormide võrdlus ning selgitatakse, miks just Arduino baasil kogu programmeeritav taevakaart valmistatud on.

Töö teises peatükis keskendutakse programmeeritava taevakaardi valguspaneeli ehitamisele. Tuuakse välja tekkinud probleemid ning nende lahendused ning soovitud probleemide ning üleliigse ajakulu ära hoidmiseks.

Antud töös püstitatud probleem sai töötava lahenduse. Autor õppis töö käigus koostama elektrisüsteeme ning rakendama nendes loogikaelemente. Samuti arendas autor oma programmeerimisoskusi ning õppis aru saama Arduino tööpõhimõtetest. Tehnilise poole pealt arendas autor antud töö käigus jootmisoskusi. Töö käigus valmistati programmeeritav taevakaardi valguspaneel. Autor ei soovita enda kogemusele baseerudes antud seadet valmistada huvilistel, kellel puudub kannatlikkus, sest töö alguse hetkest kuni valmimiseni võib kuluda erinevate tekkida võivate probleemide tõttu mitu aastat. Kindlasti on antud lõputöö suureks abiks ning aitab palju aega kokku hoida.

Käesolev taevakaardi valguspaneel sobib valmistamiseks huvilisele, kes ei karda vigu teha ning suudab mõelda mõned sammud ette, et võimalikult palju probleeme ennetada.

Töö käigus võib tekkida vajadus ehitada rakiseid, joonestada skeeme ning kogu töö algsadeks võtta ning uuesti konstrueerida. Tegemist ei ole lihtsa tööga. Töö latusamaks kulgemiseks soovitab autor lugeda Arduino ning elektroonika alaseid foorumeid, uurida sarnaste funktsioonidega seadmete valmistamisjuhendeid ning ka kõige pingelisematel hetkedel mitte alla anda.

Samas annab antud töö nii etapiti kui ka lõpptulemusena valmistajale eufoorilise eneseületusest tingitud naudingutunde. Viimane on ka peamine põhjus antud töö ettevõtmiseks ning mõistmiseks, et miski ei ole võimatu, kui endasse uskuda, ideed visualiseerida ning ennast natuke distsiplineerida ja võtta kogu protsessi kui avastusretke.

## **SUMMARY**

This paper focuses on the implementation of an Arduino Nano development platform based on LED control when creating a programmable control system for the sky map light panel. The necessary materials are explained and the problems that may arise during the work and their solutions are pointed out.

The first chapter deals with the existing literature on the topic and the working principle of an LED cube with a fairly similar structure in terms of technology. A comparison of different development platforms is also presented, and it is explained why the entire starry sky model has been made on the basis of Arduino.

The second chapter of the work focuses on building a programmable Sky Map Light panel. The problems encountered and their solutions, as well as recommendations to avoid problems and excessive time spent, are highlighted.

The problem raised in this work got a working solution. In the course of the work, the author learned to compile electrical systems and apply logic elements in them. The author also developed her programming skills and learned to understand the working principles of Arduino. On the technical side, the author developed soldering skills in the course of this work. During the work, a programmable sky map light panel was made.

Based on her own experience, the author does not recommend preparing this device for those who do not have the patience, since the work can take several years from start to finish due to various problems that may arise. Of course, this thesis is a great help and will help to save a lot of time. This sky map light panel is suitable for those who are not afraid to make mistakes and can think a few steps in advance to prevent as many problems as possible.

In the course of the work, it may be necessary to build jigs, draw diagrams and take all the work as starting parts and re-construct it. This is not an easy work. To make the work run more smoothly, the author recommends reading the Arduino and electronics forums, researching the instructions for making devices with similar functions, and not giving up even in the most stressful moments.

At the same time, this work gives the creator a feeling of euphoric self-overcoming both in process stages and as a result. The latter is also the main reason for undertaking this work and understanding that nothing is impossible if you believe in yourself, visualize the idea and discipline yourself a bit and take the whole process as a journey of discovery.

## KASUTATUD KIRJANDUSE LOETELU

1. Eesti Entsüklopeedia, SUN-TÜRG. 1996, 209
2. Chris Woodford, Resistors (2020)  
<https://www.explainthatstuff.com/resistors.html> [07.05.2021]
3. Elektroonika alused  
[https://www.hariduskeskus.ee/opiobjektid/elektrotehnika\\_alused/diod.html](https://www.hariduskeskus.ee/opiobjektid/elektrotehnika_alused/diod.html)  
[10.05.2021]
4. Veljo Sinivee, Elektroonika aabits, (2004)  
<http://parsek.yf.ttu.ee/~felc/ak/Aabits4.pdf> [07.05.2021]
5. Eesti Entsüklopeedia, SUN-TÜRG. 1996, 505
6. Chris Woodford, Transistor (2020)  
<https://www.explainthatstuff.com/howtransistorswork.html> [07.05.2021]
7. Microchip Technology Inc (2020) ATmega48A/PA/88A/PA/168A/PA/328/P  
Data Sheet  
<https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf> [10.05.2021]
8. Texas Instruments (2015) TLC5940 16-Channel LED Driver With DOT  
Correction and Grayscale PWM Control  
<https://www.ti.com/lit/ds/symlink/tlc5940.pdf> [10.05.2021]
9. Arduino (2021) Arduino UNO <https://store.arduino.cc/arduino-uno-rev3>  
[10.05.2021]
10. Arduino (2021) Arduino Nano <https://store.arduino.cc/arduino-nano>  
[10.05.2021]
11. Sparkfun (2021) Teensy <https://www.sparkfun.com/products/13736>  
[10.05.2021]
12. The Raspberry Pi Fondation <https://www.raspberrypi.org/products/raspberrypi-4-model-b/> [10.05.2021]
13. Libahunt (2016) <http://libahunt.ee/et/eka-arduino-algajatele-2016S-sisukord> [12.05.2021]
14. Arvutimaailm 4/11 (2011) <https://www.am.ee/node/2193> [12.05.2021]
15. TTÜ Robotikaklubi, Arduino (2016)  
[https://www.robotiklubi.ee/juhendid/arduino#servo\\_mootor](https://www.robotiklubi.ee/juhendid/arduino#servo_mootor) [12.05.2021]
16. TTÜ Robotikaklubi, Robot igapähele, Arduino progemine (2014)  
[https://www.robotiklubi.ee/media/kursused/robot\\_igayhele/2015/arduino\\_progemine2.pdf](https://www.robotiklubi.ee/media/kursused/robot_igayhele/2015/arduino_progemine2.pdf) [12.05.2021]

17. Instructables.com, kasutaja: MrTrick <https://www.instructables.com/Star-Map/> [12.05.2021]
18. Instructables.com, kasutaja: Djandco <https://www.instructables.com/Arduino-Star-Map/> [12.05.2021]
19. Fashion Indecor OÜ, Tähistaevas <https://www.findeco.ee/tahistaevas/> [13.05.2021]
20. Instructables.com, kasutaja: ScitechWA <https://www.instructables.com/3D-LED-Cube/> [13.05.2021]
21. Instructables.com, kasutaja: chr <https://www.instructables.com/LED-Cube-4x4x4/> [13.05.2021]
22. Asher Glick <https://aglick.com/charliecube.html> [13.05.2021]
23. Taevakaardi generaator <https://stellarium-web.org/> [20.04.2021]

## LISAD

**LISA 1 Juhtpaneeli valmistamiseks vajalikud komponendid ning nende antud projektis olnud maksumus:**

- Vineerplaat 10x1525x1525mm – **29€**, [Ehituse ABC](#)
- Data-projektor
- Saag
- Akutrell
- Puurid (2mm, 2,5mm ja 5mm)
- Must värv
- Pintsell
- Jootekolb - **19€**, [Oomipood](#)
- Jootetina – **1,50€**, [Oomipood](#)
- Kampol – **1,50€**, [Oomipood](#)
- Arduino Nano – **16,29€**, [Farnell](#)
- 3xTLC5940NT – **2,69\$**, [eBay](#)
- Makettplaat – **1,60€**, [Oomipood](#)
- Makettplaat – **5€**, [Oomipood](#)
- Makettplaat - **6€**, [Oomipood](#)
- Värvilised juhtmed (ühenduste tegemiseks) – **15€**, [Oomipood](#)
- LEDid 5mm – **0,50€/tk**, [Oomipood](#)
- NPN transistor IRF9531 – **1,03€/tk**, [Oomipood](#)
- 4,7 Kohm 0,6W takistid – **0,20€/tk**, [Oomipood](#)
- Emased ja isased pesad – **3,05€**, ebay
- Liimipüstol
- Liimipüstoli liimipulgad
- Optiline valguskaabel
- Termokahanev rüüs – **9€/6,25€**, [Oomipood/Oomipood](#)

## LISA 2 Programmikood

### ProgrammeeritavTaevakaart.ino

```
#include "Tlc5940.h" //Don't forget to download the Tlc5940 lib from
arduino playground and place it in you lib folder

#define LAYER_ONE 7 // Gate of 1. MOSFET which connects to
Layer "A0" (anodes of all Strings in Layer "A0")
#define LAYER_TWO 21 // Gate of 2. MOSFET which connects to
Layer "A1" (anodes of all Strings in Layer "A1")
#define LAYER_THREE 5 // Gate of 3. MOSFET which connects to
Layer "A2" (anodes of all Strings in Layer "A2")
#define LAYER_FOUR 22 // Gate of 4. MOSFET which connects to
Layer "A3" (anodes of all Strings in Layer "A3")

// put your setup code here, to run once:
void setup() {
  pinMode(LAYER_ONE, OUTPUT); // declare arduino nano
pin A0(LayerOne) as OUTPUT
  pinMode(LAYER_TWO, OUTPUT); // declare arduino nano
pin A1(LayerTwo) as OUTPUT
  pinMode(LAYER_THREE, OUTPUT); // declare arduino nano
pin A2(LayerThree) as OUTPUT
  pinMode(LAYER_FOUR, OUTPUT); // declare arduino nano
pin A3(LayerFour) as OUTPUT
  /* Call Tlc.init() to setup the tlc.
  You can optionally pass an initial PWM value (0 - 4095) for all
  channels.*/
  Tlc.init();
}

// put your main code here, to run repeatedly:
void loop() {
  void AllON();
  delay(1000);
  void StarsON();
  delay(1000);
  void PlanetsON();
  delay(1000);
}
```

### ProgrammeeritavTaevakaartFunksioonid.ino

```
/*
Basic Pin setup:
-----
ARDUINO TLC5940
13|-> SCLK (pin 25) OUT1 juhe "B" |1 28| OUT0 juhe "A"
12| OUT2 juhe "C" |2 27|-> GND (VPRG)
11|-> SIN (pin 26) OUT3 juhe "D" |3 26|-> SIN (pesa 11)
10|-> BLANK (pin 23) OUT4 juhe "E" |4 25|-> SCLK (pesa 13)
9|-> XLAT (pin 24) OUT5 juhe "F" |5 24|-> XLAT (pesa 9)
8| OUT6 juhe "G" |6 23|-> BLANK (pesa 10)
7| OUT7 juhe "H" |7 22|-> GND
6| OUT8 juhe "I" |8 21|-> VCC (+5V)
5| OUT9 juhe "J" |9 20|-> 2K takisti -> GND
4| OUT10 juhe "K" |10 19|-> +5V (DCPRG)
```

```

    3|-> GSCLK (pin 18)   OUT11 juhe "L" |11  18|-> GSCLK (pesa 3)
    2|                   OUT12 juhe "M" |12  17|-> SOUT (Use if U have
more then 1 TLC)
    1|                   OUT13 juhe "N" |13  16|-> XERR (veatuvastuseks,
kuid see ei ole oluline)
    0|                   OUT14 juhe "O" |14  15| OUT15 juhe "P"
-----
- Put the longer leg (anode) of the LEDs in the +5V and the shorter
leg (cathode) in OUT(0-15).
- +5V from Arduino -> TLC pin 21 and 19      (VCC and DCPRG)
- GND from Arduino -> TLC pin 22 and 27      (GND and VPRG)
- digital 3         -> TLC pin 18            (GSCLK)
- digital 9         -> TLC pin 24            (XLAT)
- digital 10        -> TLC pin 23            (BLANK)
- digital 11        -> TLC pin 26            (SIN)
- digital 13        -> TLC pin 25            (SCLK)
*/

```

```

//TLC5940 nr 1
#define AchTLC1 0 //defines the "A" column for the tlc5940 -->OUTPUT 0
#define BchTLC1 1 //defines the "B" column for the tlc5940 -->OUTPUT 1
#define CchTLC1 2 //defines the "C" column for the tlc5940 -->OUTPUT 2
#define DchTLC1 3 //defines the "D" column for the tlc5940 -->OUTPUT 3
#define EchTLC1 4 //defines the "E" column for the tlc5940 -->OUTPUT 4
#define FchTLC1 5 //defines the "F" column for the tlc5940 -->OUTPUT 5
#define GchTLC1 6 //defines the "G" column for the tlc5940 -->OUTPUT 6

```

```

//TLC5940 nr 2
#define AchTLC2 16 //defines the "A" column for the tlc5940 -->OUTPUT 0
#define BchTLC2 17 //defines the "B" column for the tlc5940 -->OUTPUT 1
#define CchTLC2 18 //defines the "C" column for the tlc5940 -->OUTPUT 2
#define DchTLC2 19 //defines the "D" column for the tlc5940 -->OUTPUT 3
#define EchTLC2 20 //defines the "E" column for the tlc5940 -->OUTPUT 4
#define FchTLC2 21 //defines the "F" column for the tlc5940 -->OUTPUT 5
#define GchTLC2 22 //defines the "G" column for the tlc5940 -->OUTPUT 6

```

```

//TLC5940 nr 3
#define AchTLC3 33 //defines the "A" column for the tlc5940 -->OUTPUT 0
#define BchTLC3 34 //defines the "B" column for the tlc5940 -->OUTPUT 1
#define CchTLC3 35 //defines the "C" column for the tlc5940 -->OUTPUT 2
#define DchTLC3 36 //defines the "D" column for the tlc5940 -->OUTPUT 3
#define EchTLC3 37 //defines the "E" column for the tlc5940 -->OUTPUT 4
#define FchTLC3 38 //defines the "F" column for the tlc5940 -->OUTPUT 5

```

```

/* ////////////////////////////////////////
////////////////////////////////////// FUNCTIONS ////////////////////////////////////////
////////////////////////////////////// */

```

```

//Turn ON all
void AllON(){
    digitalWrite(LAYER_ONE, HIGH);
    digitalWrite(LAYER_TWO, HIGH);
    digitalWrite(LAYER_THREE, HIGH);
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
}

```

```

    for (int i = 0; i<65; i++){
        Tlc.set(i, 4095);
    }
    Tlc.update();
}

//Turn ON all One by One
void AllOneByOneON() {
    digitalWrite(LAYER_ONE, HIGH);
    digitalWrite(LAYER_TWO, HIGH);
    digitalWrite(LAYER_THREE, HIGH);
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    for (int i = 0; i<58; i++){
        Tlc.set(i, 4095);
        delay(1000); //1sek=1000ms
    }
    for (int i = 59; i<65; i++){
        Tlc.set(i, 1000); //For planets less light then for stars
        delay(1000); //1sek=1000ms
    }
    Tlc.update();
}

//Turn ON all constellations and stars
void StarsON() {
    digitalWrite(LAYER_ONE, HIGH);
    digitalWrite(LAYER_TWO, HIGH);
    digitalWrite(LAYER_THREE, HIGH);
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    for (int i = 0; i<58; i++){
        Tlc.set(i, 4095);
    }
    Tlc.update();
}

//Turn ON all planets
void PlanetsON() {
    SaturnON();
    UranusON();
    SunON();
    MercuryON();
    MoonON();
    JupiterON();
    VenusON();
}

/* ////////////////////////////////////////
TURNING ON constellation, stars & planets
//////////////////////////////////////// */
// Turn ON Cassiopeia TLC1 Pin0
void CassiopeiaON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(AchTLC1, 4095);
    Tlc.update();
}

```



```

}

//Turn ON Camelopardalis TLC1 Pin1
void CamelopardalisON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(BchTLC1, 4095);
    Tlc.update();
}

//Turn ON Perseus TLC1 Pin2
void PerseusON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(CchTLC1, 4095);
    Tlc.update();
}

//Turn ON Trianglum TLC1 Pin3
void TrianglumON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(DchTLC1, 4095);
    Tlc.update();
}

//Turn ON Andromeda TLC1 Pin4
void AndromedaON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(EchTLC1, 4095);
    Tlc.update();
}

//Turn ON Pegasus TLC1 Pin5
void PegasusON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(FchTLC1, 4095);
    Tlc.update();
}

//Turn ON Lancerta TLC1 Pin0
void LancertaON() {
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(AchTLC1, 4095);
    Tlc.update();
}

//Turn ON Cepheus TLC1 Pin1
void CepheusON() {
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(BchTLC1, 4095);
    Tlc.update();
}

```

```

//Turn ON Polaris (Ursa Minor) TLC1 Pin2
void PolarisON() {
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(CchTLC1, 4095);
    Tlc.update();
}

//Turn ON Ursa Minor TLC1 Pin2+Pin3
void UrsaMinorON() {
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(CchTLC1, 4095);
    Tlc.set(DchTLC1, 4095);
    Tlc.update();
}

//Turn ON Draco TLC1 Pin4
void DracoON() {
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(EchTLC1, 4095);
    Tlc.update();
}

//Turn ON Cygnus TLC1 Pin5
void CygnusON() {
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(FchTLC1, 4095);
    Tlc.update();
}

//Turn ON Pisces TLC1 Pin0
void PiscesON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(AchTLC1, 4095);
    Tlc.update();
}

//Turn ON Ancient (Aries) TLC1 Pin1
void AncientON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(BchTLC1, 4095);
    Tlc.update();
}

//Turn ON Aries TLC1 Pin1+Pin2
void AriesON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(BchTLC1, 4095);
    Tlc.set(CchTLC1, 4095);
    Tlc.update();
}

```

```

}

//Turn ON Pleiades (Taurus) TLC1 Pin3
void PleiadesON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(DchTLC1, 4095);
    Tlc.update();
}

//Turn ON Aldebaran (Taurus) TLC1 Pin4
void AldebaranON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(EchTLC1, 4095);
    Tlc.update();
}

//Turn ON Taurus TLC1 Pin3+Pin4+Pin5
void TaurusON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(DchTLC1, 4095);
    Tlc.set(EchTLC1, 4095);
    Tlc.set(FchTLC1, 4095);
    Tlc.update();
}

//Turn ON Auriga TLC1 Pin1
void AurigaON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(BchTLC1, 4095);
    Tlc.update();
}

//Turn ON Lynx TLC1 Pin2
void LynxON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(CchTLC1, 4095);
    Tlc.update();
}

//Turn ON Ursa Major TLC1 Pin3
void UrsaMajorON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(DchTLC1, 4095);
    Tlc.update();
}

//Turn ON Ursa Major Complete TLC1 Pin3+Pin4
void UrsaMajorCompleteON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(DchTLC1, 4095);
}

```

```

    Tlc.set(EchTLC1, 4095);
    Tlc.update();
}

//Turn ON Hercules TLC1 Pin5
void HerculesON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(FchTLC1, 4095);
    Tlc.update();
}

//Turn ON Vega(Lyra) TLC2 Pin0
void VegaON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(AchTLC2, 4095);
    Tlc.update();
}

//Turn ON Lyra TLC1 Pin0+Pin1
void LyraON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(AchTLC2, 4095);
    Tlc.set(BchTLC2, 4095);
    Tlc.update();
}

//Turn ON Delphinus TLC2 Pin2
void DelphinusON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(CchTLC2, 4095);
    Tlc.update();
}

//Turn ON Equuleus TLC2 Pin3
void EquuleusON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(DchTLC2, 4095);
    Tlc.update();
}

//Turn ON Altair (Aquila) TLC2 Pin4
void AltairON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(EchTLC2, 4095);
    Tlc.update();
}

//Turn ON Aquila TLC2 Pin4+Pin5
void AquilaON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
}

```

```

    Tlc.set(EchTLC2, 4095);
    Tlc.set(FchTLC2, 4095);
    Tlc.update();
}

//Turn ON Aquarius TLC2 Pin0
void AquariusON(){
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(AchTLC2, 4095);
    Tlc.update();
}

//Turn ON Cetus TLC2 Pin1
void CetusON(){
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(BchTLC2, 4095);
    Tlc.update();
}

//Turn ON Orion Belt TLC2 Pin2
void OrionBeltON(){
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(CchTLC2, 4095);
    Tlc.update();
}

//Turn ON Rigel (Orion) TLC2 Pin3
void RigelON(){
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(DchTLC3, 4095);
    Tlc.update();
}

//Turn ON Betelgeuse (Orion) TLC2 Pin4
void BetelgeuseON(){
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(EchTLC3, 4095);
    Tlc.update();
}

//Turn ON Orion TLC2 Pin2+Pin3+Pin4+Pin5
void OrionON(){
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(CchTLC2, 4095); //Orion Belt
    Tlc.set(DchTLC2, 4095); //Rigel (Orion)
    Tlc.set(EchTLC2, 4095); //Betelgeuse (Orion)
    Tlc.set(FchTLC2, 4095); //Orion
    Tlc.update();
}

//Turn ON Eridanus TLC2 Pin1

```

```

void EridanusON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(BchTLC2, 4095);
    Tlc.update();
}

//Turn ON Gemeni TLC2 Pin2
void GemeniON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(CchTLC2, 4095);
    Tlc.update();
}

//Turn ON Cancer TLC2 Pin3
void CancerON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(DchTLC2, 4095);
    Tlc.update();
}

//Turn ON Leo Minor TLC2 Pin4
void LeoMinorON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(EchTLC2, 4095);
    Tlc.update();
}

//Turn ON Canes Venatici TLC2 Pin5
void CanesVenaticiON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(FchTLC2, 4095);
    Tlc.update();
}

//Turn ON Arcturus (Bootes) TLC2 Pin1
void ArcturusON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(BchTLC2, 4095);
    Tlc.update();
}

//Turn ON Bootes TLC2 Pin1+Pin2
void BootesON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(BchTLC2, 4095);
    Tlc.set(CchTLC2, 4095);
    Tlc.update();
}

//Turn ON Corona Borealis TLC2 Pin3

```

```

void CoronaBorealisON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(DchTLC2, 4095);
    Tlc.update();
}

//Turn ON Serpens Caput TLC2 Pin4
void SerpensCaputON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(EchTLC2, 4095);
    Tlc.update();
}

//Turn ON Ophinchus TLC2 Pin5
void OphinchusON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(FchTLC2, 4095);
    Tlc.update();
}

//Turn ON Serpens Cauda TLC3 Pin0
void SerpensCaudaON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(AchTLC3, 4095);
    Tlc.update();
}

//Turn ON Scutum TLC3 Pin1
void ScutumON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(BchTLC3, 4095);
    Tlc.update();
}

//Turn ON Capricornus TLC3 Pin2
void CapricornusON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(CchTLC3, 4095);
    Tlc.update();
}

//Turn ON Fomalhaut (Piscis Austrinus) TLC3 Pin3
void FomalhautON() {
    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(DchTLC3, 4095);
    Tlc.update();
}

//Turn ON Piscis Austrinus TLC3 Pin3+Pin4
void PiscisAustrinusON() {

```

```

    digitalWrite(LAYER_ONE, HIGH);
    Tlc.clear();
    Tlc.set(DchTLC3, 4095);
    Tlc.set(EchTLC3, 4095);
    Tlc.update();
}

//Turn ON Sculptor TLC3 Pin0
void SculptorON() {
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(AchTLC3, 4095);
    Tlc.update();
}

//Turn ON Grus TLC3 Pin1
void GrusON() {
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(BchTLC3, 4095);
    Tlc.update();
}

//Turn ON Microscopium TLC3 Pin2
void MicroscopiumON() {
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(CchTLC3, 4095);
    Tlc.update();
}

//Turn ON Indus TLC3 Pin3
void IndusON() {
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(DchTLC3, 4095);
    Tlc.update();
}

//Turn ON Sagittarius TLC3 Pin4
void SagittariusON() {
    digitalWrite(LAYER_TWO, HIGH);
    Tlc.clear();
    Tlc.set(EchTLC3, 4095);
    Tlc.update();
}

//Turn ON Corona Australis TLC3 Pin0
void CoronaAustralisON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(AchTLC3, 4095);
    Tlc.update();
}

//Turn ON Scorpius TLC3 Pin1
void ScorpiusON() {

```



```

    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(BchTLC3, 4095);
    Tlc.update();
}

//Turn ON Libra TLC3 Pin2
void LibraON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(CchTLC3, 4095);
    Tlc.update();
}

/* //////////////////////////////////////
   ////////// TURNING ON THE PLANETS //////////
   ////////////////////////////////////// */

//Turn ON Saturn TLC3 Pin3
void SaturnON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(DchTLC3, 1000); //For planets less light then for stars
    Tlc.update();
}

//Turn ON Uranus TLC3 Pin4
void UranusON() {
    digitalWrite(LAYER_THREE, HIGH);
    Tlc.clear();
    Tlc.set(EchTLC3, 1000); //For planets less light then for stars
    Tlc.update();
}

//Turn ON Sun TLC3 Pin0
void SunON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(AchTLC3, 1000); //For planets less light then for stars
    Tlc.update();
}

//Turn ON Mercury TLC3 Pin1
void MercuryON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(BchTLC3, 1000); //For planets less light then for stars
    Tlc.update();
}

//Turn ON Moon TLC3 Pin2
void MoonON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(CchTLC3, 1000); //For planets less light then for stars
    Tlc.update();
}

```

```
//Turn ON Jupiter TLC3 Pin3
void JupiterON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(DchTLC3, 1000); //For planets less light then for stars
    Tlc.update();
}

//Turn ON Venus TLC3 Pin4
void VenusON() {
    digitalWrite(LAYER_FOUR, HIGH);
    Tlc.clear();
    Tlc.set(EchTLC3, 1000); //For planets less light then for stars
    Tlc.update();
}
```