

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Robert Arus 185082IADB

# **Nõelmaatriksprinterit juhtprogrammi loomine, kasutades Arduino mikrokontrollerit**

Bakalaureusetöö

Juhendaja: Priit Ruberg  
PhD

Tallinn 2023

## **Autorideklaratsioon**

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Robert Arus

15.05.2023

## **Abstract**

### **Developing a Dot Matrix Printer Driver Using an Arduino Microcontroller**

The thesis is written in Estonian and contains 26 pages of text and 6 chapters. The aim of the thesis is to create a driver for an Epson FX-80 dot matrix printer using an Arduino microcontroller. The driver will be used in a Windows 10 environment to allow the user of the application to input text on a graphical user interface in a Windows application and print that text out with the Epson printer. The graphical user interface will be a Windows Forms desktop application written in C# language. The Arduino will be programmed in the Arduino IDE using both C++ and Arduino syntax (which is based on C++). The controller used in the project is an Arduino MEGA 2560.

## Lühendite ja mõistete sõnastik

Arduino IDE	Arduino platvormi tarkvaraarenduskeskkond
BAUD	Modulatsioonikiirus, bitti sekundis
BUSY	Hõivatuse signaal
CR	<i>Carriage Return</i> , tagasijooks, prindipea naasmine rea algusesse
Crosstalk	Läbikoste ehk signaali soovimatu ülekandumine ühest sidekanalist teise
FF	<i>Form Feed</i> , lehevahetus, printimise jätkamine järgmisel lehel
GPIO	Mitmeotstarbeline sisend/väljundviik
LF	<i>Line Feed</i> , reavahetus, paberi nihutamine prindipea ees rea võrra edasi
NLQ	<i>Near Letter Quality</i> , lihtkvaliteet, prindi kvaliteediklass
LQ	<i>Letter Quality</i> , esinduskvaliteet, maatriksprinterite kõrgeim kvaliteediklass
PWM	<i>Pulse-width modulation</i> ehk impulsilaiusmodulatsioon on modulatsiooni liik, milles väljundpinge reguleerimiseks muudetakse impulsside laiust
ROM	<i>Read-only memory</i> . Püsimälu, ainult loetav mälu
STROBE	Andmevahetustsükli tunnusimpulsi signaal
USB	Universaalne järjestiksiin

## Sisukord

1 Sissejuhatus .....	6
2 Taustuuring .....	7
2.1 Mängulisusest õppimisel .....	7
2.2 Nõelmaatriksprinterite ajaloost.....	8
2.3 Epson FX-80.....	10
2.4 Andmevahetus printeriga.....	10
2.5 Nõelmaatriksprinterid ja kaasaegne operatsioonisüsteem .....	13
2.6 Mikrokontrolleritest.....	13
2.7 Arduino MEGA 2560 .....	15
3 Lahenduskäik.....	16
3.1 Printeri seadistamine.....	16
3.2 Printeri juhtpaneel.....	17
3.3 Lindi värskendamine .....	17
3.4 Töökorrasoleku kontroll .....	18
3.5 Riistvara ülesseadmine .....	19
3.6 Ühendusklemmide ühendamine Arduino külge .....	19
3.7 Võimalikud edasiarendused.....	21
4 Tarkvara disain .....	21
5 Kokkuvõte .....	23
Kasutatud materjalid.....	24
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	27
Lisa 2 – Graafiline kasutajaliides ja selle kood .....	28
Lisa 3 – Arduino IDE-s kirjutatud programmi kood .....	30
Lisa 4 – Kirjavahetus Vimeo kasutaja MIDIDesaster-iga.....	34

# 1 Sissejuhatus

Modernne maailm põhineb tehnoloogial. Tehnoloogia areng ja levik pea igasse mõeldavasse valdkonda on muutnud põhjalikult viise, kuidas me maailmaga suhtleme ja suhestume. Nutitelefonid, arvutid ja internet on saanud meie maailmapildi lahutamatuks osaks. Tänu nendele vahenditele on meil võimalik suhelda ja jagada oma elus toimuvat sõprade ja tuttavatega üle kogu maailma, leida rikkalikult mitmekesisist teavet hetkega, jäädvustada ennast ümbritsevat keskkonda enneolematu kvaliteediga ning teha igapäevatööd efektiivsemalt. Hiljuti tormiliselt inimeste teadvuse vallutanud keelemudel-tüüpi juturobot Chat GPT lisab veel enam integreeritust modernse maailma ja tehnoloogia vahele.

Elektroonika ja tarkvaraarenduse õppimine on eelneva tõttu muutunud ühiskonnas aina olulisemaks. Tehnoloogia arengu, leviku ja haldamise elluviimise eelduseks on pädevate teadmiste ja vastavate praktiliste kogemustega insenerid. Need teadmised ja oskused avavad ukse kaasaegsele maailmale, kuid nende ainetega tegelemine võib olla paljudele õpilastele keeruline väljakutse. Nii elektroonika kui tarkvaraarendus on kognitiivselt paljunõudvad valdkonnad ning kui õppetöö muutub igavaks ja ebaatraktiivseks, on huvi kadumine kerge tulema. Seetõttu on oluline luua huvitavaid praktilisi ülesandeid, mis aitavad õpilastel neid valdkondi süvitsi õppida.

Käesoleva lõputöö eesmärk on luua juhtprogramm nõelmaatriksprinteriga Epson FX-80 ühendamiseks Arduino mikrokontrolleriga. Juhtprogrammiga on võimalik printeriga trükkida paberile graafilisel kasutajaliidesel ette trükitud tekst ning luua eelseadistatud graafilisi kujutisi. Lõputöö saab juhiseks samaväärse projekti või selle edasiarenduse elluviimiseks elektroonikast ja arvutiteadusest huvitatud õpilastel või tudengitel.

## 2 Taustuuring

### 2.1 Mängulisusest õppimisel

Kaasaegse maailma sõltuvus tehnoloogiast on ühteaegu nii positiivne kui negatiivne nähtus. Ühelt poolt pole info kättesaadavus kunagi olnud nii vahetu ja mitmekesine, teisalt pole harv nähtus, et sellest infost ja oma tehnoloogiaseadmetest võib tekkida sõltuvus.

Veel enam olulisem on kaasajal rakendada kõiki võimalikke meetmeid, et teha õppimine huvitavaks ja atraktiivseks. Strateegiaid selleks on mitmeid.

Oma uurimustöös [1] võrdlesid Lissaboni ülikooli teadurid tagajärgi, kui nende ülikoolis õpetatav magistriõppekava õppeaine „Multimeedia sisu loomine“ tehti mängulisemaks. Õppeaine kavasse lisati enamjaolt arvutimängudest tuntud aspekte: kogemuspunktid (*XP*), edenemistasemed, avalik pidevalt uuenev edetabel, väljakutsed ja märgid erisaavutuste eest.

Uurimistöö tulemused näitasid, et mängulisem õppeaine ülesehitus parandas üliõpilaste aktiivsust, motiveeris neid suhtlema kursuse jooksul aktiivsemalt ning parandas loengutes tudengite tähelepanelikkust.

Veel üks võimalik strateegia on luua projekte, mis on atraktiivsed oma olemuselt. Kas lahendatakse omapäraselt mõni olemasolev probleem või kasutatakse mõnda seadet või nähtust omapärasel viisil. Näitena sellistest projektidest on erinevatel elektroonilistel seadmetel muusika mängimine ning üks sellise projekti väga populaarne näide on Floppotron. [2]

Selle looja on Poola tarkvarainsener Paweł Zadrożniak ning tema esimene Floppotronist tehtud video kogus 2011. aastal 5 päevaga üle 2 miljoni vaatamise [3]. Kõige viimatisem projekti edasiarendus - Floppotron 3.0 - koosneb 512-st disketiseadmest, 4 skannerist ning 16st kõvakettast [4].

Veel väga populaarseid sarnaseid hobiprojekte on loonud videojagamisplatvormi Vimeo kasutaja MIDIDesaster [5], kes kasutab heli loomiseks nõelmaatriksprintereid ning

Youtube-i kasutaja Device Orchestra: [6] kes loob muusikat kasutades erinevaid kaardimakse seadmeid, elektrilisi hambaharju ja kirjutusmasinaid.

Lõputöö autor usub, et sellised, mitmete miljonite vaatamistega projektid, tekitavad noortes õpilastes ja tudengites iseenesest huvi ning nende põhjal on võimalik luua atraktiivseid ja huvitavaid õppetöös kasutatavaid projekte. Sel põhjusel sai käesoleva lõputöö elluviimiseks valitud just juhtprogrammi loomine nõelmaatriksprinterile.

Nõelmaatriksprinter sai lisaks eelnevale valitud antud lõputöö elluviimiseks järgnevatel põhjustel:

- Nõelmaatriksprinteriga on võimalik viia ellu nii lihtsaid kui ka kompleksseid projekte.
- Tulemus on näha koheselt: kas ja mida printer paberile prindib.
- Kõik nõelmaatriksprinteriga seotud projektid nõuavad teadmisi (või nende omandamist) mitmest erinevast valdkonnast: elektroonika, programmeerimine, andmeside, jt.
- Printeriga saab ellu viia efektseid projekte. Näiteks printeri abil graafika või heli loomine.

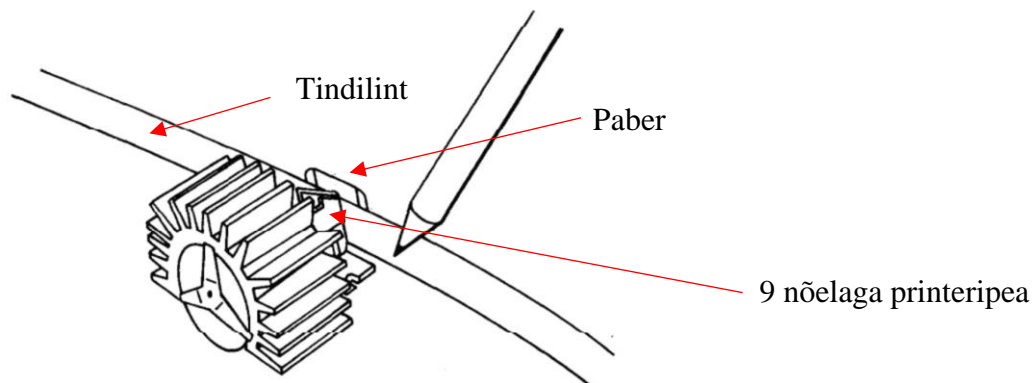
## **2.2 Nõelmaatriksprinterite ajaloost**

Printimine on mänginud olulist rolli infoühiskonna arengus. Tehnoloogiaid, mis võimaldavad digitaalselt salvestatud tähemärke, tekste ja graafiliseid kujutiseid talletada füüsilistele andmekandjatele, on arendatud sama kaua, kui on arendatud salvestust võimaldavaid arvuteid. Esimesed printerid võeti kasutusele elektronarvutustehnika algusaastatel. Näiteks 1951.a. valminud elektronarvutis „Univac“ kasutati spetsiaalset „Uniprinteri“ nimelist reaprinterit, mis suutis trükkida kiirusega kuni 600 märki sekundis [7]. Printimistehnoloogiad saab tööpõhimõtte järgi kategoriseerida kahte suurde gruppi: löögiga ja löögita printerid.

Elektromehaaniline löökprintimine on ajalooliselt kõige vanem prinditehnoloogia, kuid selle osatähtsus on kaasajal vähenenud miinimumini. Peamisteks põhjusteks on löökprintimisega kaasnev müra ning alternatiividega võrreldes kehvem printimiskvaliteet. Eeliste osas saab välja tuua löökprintimise töökindluse ja pika eluea.

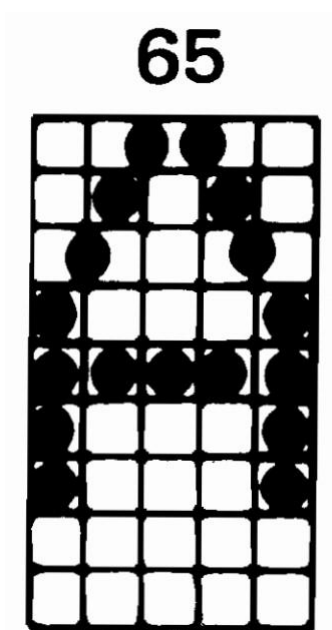


Elektromehaanilise löökprintimise saab omakorda jagada nõel-, õis- ja ridaprintimiseks. Nõelprintimise tööpõhimõte on lihtne: printeripeas paiknevad nõelad löövad tindist läbi immutatud linti, mille tulemusel kandub tint lindi all olevale paberile [7].



Joonis 1. Printeripea ja tindilint

Üheks löökprinterite (sh nõelmaatriksprinterite) miinuseks on teiste tehnoloogiatega võrreldes oluliselt suurem müranivoo, mis löökprinteirtel ulatub 60-70 akustilise detsibellini. Võrdluseks on laserprinterite müranivoo printimise ajal 45-50 dB(A). [7] Nõelprinteri nõelad paiknevad üldjuhul kohakuti üksteise peal, mistõttu on võimalik ühe löögiga paberile kanda ainult ühe nõela laiune vertikaalne rida.



Joonis 2. Tähemärgi 'A' trükkimine 9 nõelaga maatriksprinteriga [11]

Joonisel 2. on näha tähemärgi 'A' trükkimine 9-nöelaga nõelprinteriga, kus kõik nõelte löögid on eristatavad.

Tavaliste nõelprinterite väljastuskiirus on suurusjärgus 200-300 märki/sekundis [7]. Käesolevas diplomitöös kasutatav nõelmaatriksprinter Epson FX-80 suudab trükkida kuni 160 tähemärki sekundis. Iga tähemärk trükitakse 9 x 11 maatriksile (s.h. eelnev ja järgnev tühi ala) ning keskmine tähemärgi laius on 6 pikslit.

### **2.3 Epson FX-80**

Käesolevas diplomitöös kasutatakse nõelmaatriksprinterit Epson FX-80 (seerianumbriga 528550), mis on Jaapani firma *Seiko Epson* toodang. FX-80 on järeltulija Epsoni MX-80 nõelmaatriksprinterile ning nende peamine erinevus seisneb selles, et FX-80-le on lisatud 2 KB puhvermälu, tänu millele on kasutajal võimalus ise prinditavaid tähemärke kujundada [7].

Printeri toitekaabel on standardne C13 tüüpi toitekaabel, mis töötab karakteristikutega 220V, 50/60 Hz.

Andmevahetus printeriga on võimalik nii seeria- kui rööpülekande kaudu, kuid seeriaülekande jaoks on vaja soetada juurde eraldiseisev seeriaülekande kontrolleri.

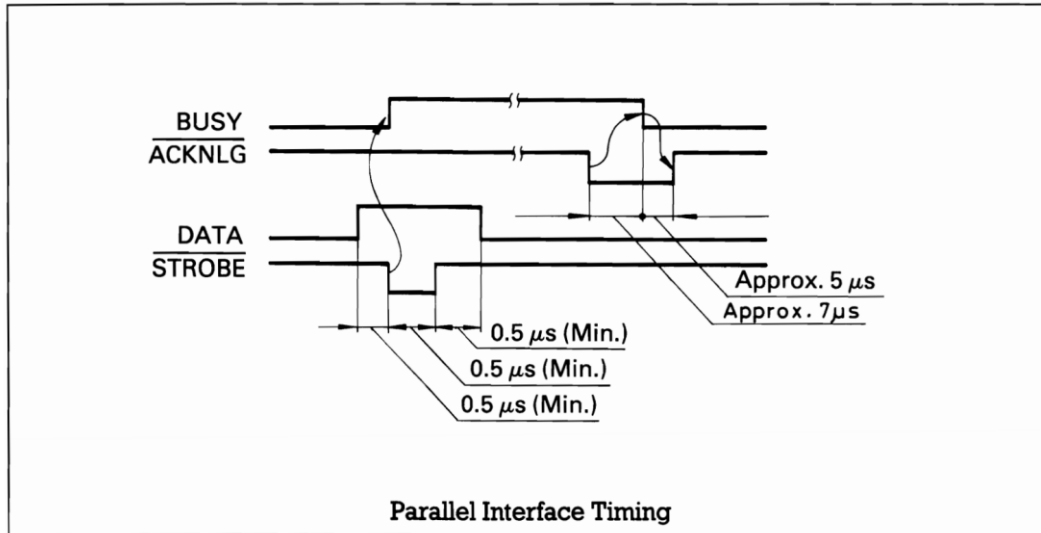
Rööpülekande liides on standardne *Centronics*-tüüpi 36 klemmiga pistik.

### **2.4 Andmevahetus printeriga**

Andmevahetus printeriga on võimalik nii seeria- kui rööpülekande kaudu. Projektis kasutataval printeril on sisseehitatud *Centronics*-tüüpi 36-klemmiga rööpühenduse pistik. Nimi *Centronics* tuleneb Ameerika Ühendriikide ettevõtte Centronics Corporation nimest, kelle pistikud said 1970ndatel *de facto* standardiks mitme erineva tootja printeritel [8].

Epson FX-80-ga on võimalik suhelda ka seeriaülekande kaudu, kuid selleks on vaja soetada lisaks eraldiseisev seeriaülekande kontrolleri *#8148 Intelligent Serial Interface*. [9].

Küsimusele, kas see on antud projektis vajalik ja/või otstarbekas, saab vastuse analüüsid, kas olemasolevast rööpülekande ühendusest meile projekti teostamiseks piisab. Selleks toome välja signaalide nõutava järjestuse graafiku Joonisel 3 [11] ja kasutusjuhendi [7] lisas C lehekülgedel C-1 ja C-2 väljatoodud signaaliklemmide tabelid Tabel 1 ja 2.



Joonis 3. Signaalide järjestus [11]

**Table C-1 Connector Pin Assignment and Descriptions of Interface Signals**

Signal Pin No.	Return Pin No.	Signal	Direction	Description
1	19	STROBE	In	STROBE pulse to read data in. Pulse width must be more than 0.5 $\mu$ s at receiving terminal.
2	20	DATA 1	In	These signals represent information of the 1st to 8th bits of parallel data respectively. Each signal is at "HIGH" level when data is logical "1" and "LOW" when logical "0".
3	21	DATA 2	In	
4	22	DATA 3	In	
5	23	DATA 4	In	
6	24	DATA 5	In	
7	25	DATA 6	In	
8	26	DATA 7	In	
9	27	DATA 8	In	
10	28	ACKNLG	Out	Approx. 12 $\mu$ s pulse. "LOW" indicates that data has been received and that the printer is ready to accept other data.
11	29	BUSY	Out	A "HIGH" signal indicates that the printer cannot receive data. The signal becomes "HIGH" in the following cases: 1. During data entry 2. During printing operation 3. In OFF-LINE state 4. During printer error status.

Tabel 1. Rööpliidese printeripoolsete ühenduste tabel (1/2)

Signal Pin No.	Return Pin No.	Signal	Direction	Description
12	30	PE	Out	A "HIGH" signal indicates that the printer is out of paper.
13	—	—	—	Pulled up to +5V through 3.3 k $\Omega$ resistance
14	—	AUTO FEED XT	In	With this signal being at "LOW" level, the paper is automatically fed one line after printing. (The signal level can be fixed to "LOW" with DIP SW pin 2-4 provided on the control circuit board.)
15	—	NC	—	Not used.
16	—	0V	—	Logic GND level.
17	—	CHASSIS GND	—	Printer chassis GND. In the printer, the chassis GND and the logic GND are isolated from each other.
18	—	NC	—	Not used.
19 to 30	—	GND	—	TWISTED-PAIR RETURN signal GND level.
31	—	INIT	In	When the level of this signal becomes "LOW", the printer controller is reset to its initial state and the print buffer is cleared. This signal is normally at "HIGH" level, and its pulse width must be more than 50 $\mu$ s at the receiving terminal.
32	—	ERROR	Out	The level of this signal becomes "LOW" when the printer is in— 1. PAPER END state 2. OFF-LINE state 3. Error state
33	—	GND	—	Same as with Pin Nos. 19 to 30.
34	—	NC	—	Not used.
35	—	—	—	Pulled up to +5V through 3.3 k $\Omega$ resistance.
36	—	SLCT IN	In	The DC1/DC3 code is only valid when this signal is "HIGH" level. (Internal fixing can be carried out with DIP SW pin 2-1. The level of this signal is factory-set to "LOW".)

Tabel 2. Rööpliidese printeripoolsete ühenduste tabel (2/2)

Signaalide järjestuse graafikust järeldub, et andmete edastamiseks tuleb esialgu oodata kuni hõivatuse (*BUSY*) signaali nivoo on madalal, misjärel on võimalik andmesiinile edastada andmete bitid. Seejärel tuleb oodata vähemalt 0,5 mikrosekundit, et saaks andmevahetustsükli tunnusimpulsi loogiliselt kõrgele nivoole tõstmisega printerile teada anda, et andmed on valmis edastamiseks.

Kui tunnusimpulsi nivoo on kõrgele ning vajaminev viiteaeg on möödunud, tuleb andmete edastamiseks tunnusimpulss langetada loogiliselt madalale nivoole. Seejärel on andmed printerile edastatud ning järgmiste andmete edastamiseks on vaja oodata kuni printeri hõivatuse signaali jälle madalale nivoole langetab.

Printeri taaskäivitamise (*INIT*) signaal tühjendab andmesiinil olevad bitid ning taastab printeri enda kontrolleri tavaoleku ehk kõik võimalikud kontrolleri saadetud teksti stiili käsud (näiteks kursiiv) tühistatakse.

## **2.5 Nõelmaatriksprinterid ja kaasaegne operatsioonisüsteem**

Vanemate nõelmaatriksprinterite ühendamine kaasaegsete operatsioonisüsteemidega on võimalik, kuid muutub aastast aastas keerulisemaks, jäädes pigem hobielektroonikute ja arvutientusiastide pärusmaaks. Protokollid ja andmevahetusstandardid, mida nõelmaatriksprinterid kasutavad, ei leia kasutust tänapäevastes enimlevinud tindi- ja laserprinterites ning liidesed ja pistikud, mida kasutavad vanemad nõelmaatriksprinterid, ei ühildu enimlevinud kaablite abil kaasaegsete arvutitega.

Olemasolevatet meetoditest on üks variant soetada erikaabel [12], mis ühendab omavahel arvutipoolse USB pistiku ning printeripoolse rööp-pistiku. Sellise meetodi kasutamine nõuab ka operatsioonisüsteemile sobiva printeri juhtprogrammi alla laadimist. Epsoni kasutatav nõelmaatriksprinterite andmevahetusstandard ESC/P on laialtlevinud ning seda kasutavad ka mitme teise tootja nõelmaatriksprinterid [13]. Ehk ülesanne leida printerile juhtprogramm taandub vähemalt ühe 9-nõelaga maatriksprinter Windows 10 juhtprogrammi leidmisele, mis kasutab sama andmevahetusstandardit. Jaapani firma OKI [14] printeritele on selline juhtprogramm saadaval [15].

Käesoleva lõputöö lahendus on taotluslikult valitud selliselt, kus operatsioonisüsteemist saadetakse läbi käsud ja andmed printerile läbi vahepealse mikrokontrolleri. Printer on mikrokontrolleriga ühendatud rööppistiku kaudu.

## **2.6 Mikrokontrolleritest**

Antud töös lahendatava ülesande jaoks oli vaja valida sobiv mikrokontroller. Tehnilise poole pealt oli nõutav piisav sisendite ja väljundite arv, kontrolleri ja nõelmaatriksprinter pingeniivoode kattuvus ning vajalike ühenduskomponentide olemasolu.

Kui tehnilise poole tingimused olid rahuldatud, prioritseeris töö autor kontrolleri ja selle taga oleva platvormi lihtsust ja kasutajasõbralikkust, kontrolleri ja selle komponentide kättesaadavust Eestist ning üldist tuntust ja populaarsust maailmas, mis loob eeldused

selleks, et platvormi jaoks on loodud laialdaselt erinevaid teeke, mida vajadusel saab kasutada ning aitab vajadusel rahvusvaheliselt kogukonnalt nõu ja abi küsida.

Valik langes Arduino kontrolleri ja platvormile.

Arduino platvormi plussid:

- Arduino on isehäälestuv (*plug and play*).
- Arduinol on võimas arenduskeskkond
- Arduino programmeerimiskeel on lihtsustatud versioon C++-ist, mida on võimalik ka algajatel lühikese ajaga selgeks saada.
- Arduino kontrolleri ja nende mõeldud lisad on odavad ja kergelt kättesaadavad.
- Arduino platvorm on avatud lähtekoodiga.
- Toetav kogukond sh avalik populaarne foorum küsimuste lahendamiseks.

Arduino platvormi miinused:

- Arduino foorum on ingliskeelne ning väga häid jooksva abi saamiseks mõeldud kohalikke allikaid ei ole
- Arduino on peamiselt mõeldud kasutamiseks C++ keelega. Teegid teiste keelte jaoks on haruldased ja nõuavad suuremaid teadmiseid ja oskuseid.
- Arduino kontrolleri jaoks saab laadida ja selle peal jooksutada ühte programmi korraga. Komplekssemate projektide korral võib selline piirang tekitada puudujääke.

[16].

Tõestus Arduino kogukonna aktiivsuse kohta on leitav nende ametliku foorumi statistikast Arduino foorumi kohta: viimase 30 päeva jooksul (07.05 seisuga) on foorumisse tehtud 51 800 uut postitust ja aktiivselt on foorumit kasutanud 13 400 inimest [17].

Kaalutav alternatiiv on võrdväärset Arduinole suurepärase kättesaadavusega Raspberry Pi Pico mikrokontrolleriplaat [18]. Meile oluliste näitajate osas on Pico 26 multifunktsionaalset GPIO viiku ja USB ühenduse olemasolu. Mõnevõrra ebamugavam on Raspberry Pi Pico plaadile tarkvara laadimine, mida tuleb teha kasutajal endal. Suure eelis Raspberry Pico on selle hind: 01.05.2023 seisuga oomipood.ee veebipoes 11 EUR (Pico) vs 61.49 EUR (Mega 2560).

## **2.7 Arduino MEGA 2560**

Konkreetselt mikrokontrolleriks valiti Arduino MEGA 2560 [19].

- Põhineb Arduinol.
- Kättesaadav. Oli võimalik osta mitmest erinevast e-poest Eestis.
- Piisav arv sisendeid ja väljundeid.

## 3 Lahenduskäik

### 3.1 Printeri seadistamine

Käesolevas diplomitöös kasutatakse nõelmaatriksprinterit Epson FX-80 (seerianumbriga 528550), mis on jaapani firma *Seiko Epson* toodang. FX-80 on järeltulija Epsoni MX-80 nõelmaatriksprinterile ning nende peamine erinevus seisneb selles, et FX-80-le on lisatud 2 KB puhvermälu, tänu millele on kasutajal võimalus ise prinditavaid tähemärke kujundada [7].

Printeri toitekaabel on standardne C13 tüüpi toitekaabel, mis töötab karakteristikutega 220V, 50/60 Hz.

Andmevahetus printeriga on võimalik nii seeria- kui rööpülekande kaudu, kuid seeriaülekande jaoks on vaja soetada juurde juba eelnevas peatükis mainitud eraldiseisev seeriaülekande kontrolleri. Eelnevas peatükis toimunud arutluse tulemusel eraldi sellise kontrolleri soetamine otstarbekas ja vajalik ei ole.

Rööpülekande liides on standardne *Centronics*-tüüpi 36 klemmiga pistik.

Elektrooniline seade vajab reegli järgi kahte kaablit: toitekaablit ja andmevahetuskaablit. Töös kasutataval printeril Epson FX-80 oli toitekaabel (standardne C13 tüüpi kaabel) kaasas, kuid puudus kaabel andmevahetuseks.

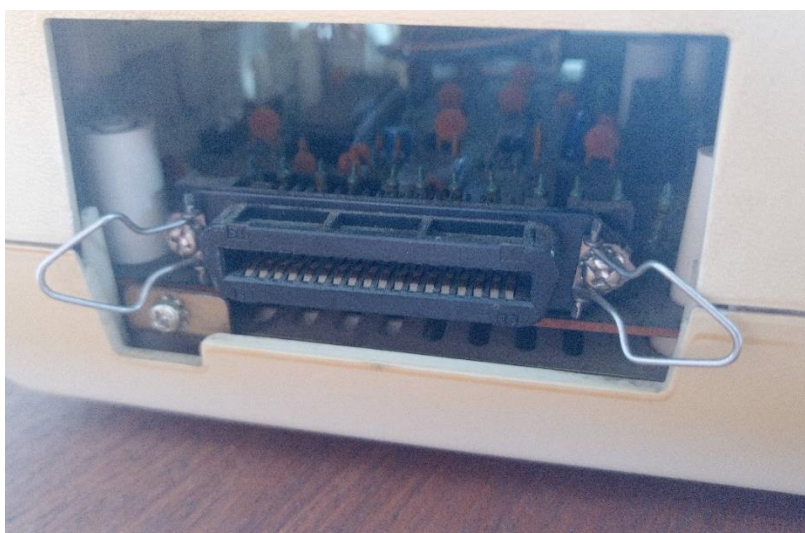


Foto 1. Printeripoolne Centronics pistikupesa



### 3.2 Printeri juhtpaneel



Foto 2. Epson FX-80 juhtpaneel

Printeri pealmisel küljel asub juhtpaneel, mis koosneb järgnevatest osadest: ON LINE, FF ja LF nupud; POWER, READY, PAPER OUT ja ON LINE tulukesed.

Kui juhtpaneeli ON LINE tuli põleb, siis printer ja arvuti on omavahel otseselt ühenduses ning FF ja LF nuppe kasutada pole võimalik. FF ja LF nuppude kasutamiseks tuleb vajutada ON LINE nuppu, et arvuti ja printeri ühendus katkestada. [20]

### 3.3 Lindi värskendamine

Vanemate nõelmaatriksprinterite levinud probleem seisneb tindilindi kuivamises, mistõttu printida pole kas võimalik või on prinditud tekst vähekontrastne. Ühe võimaliku lahendusena on vanemaid tindilinte edukalt leotatud tuntud universaalmäärdes WD-40. [20]. Määre töötab kui lahusti, aidates kuivanud tindil ühtlaselt üle lindi laiali valguda.

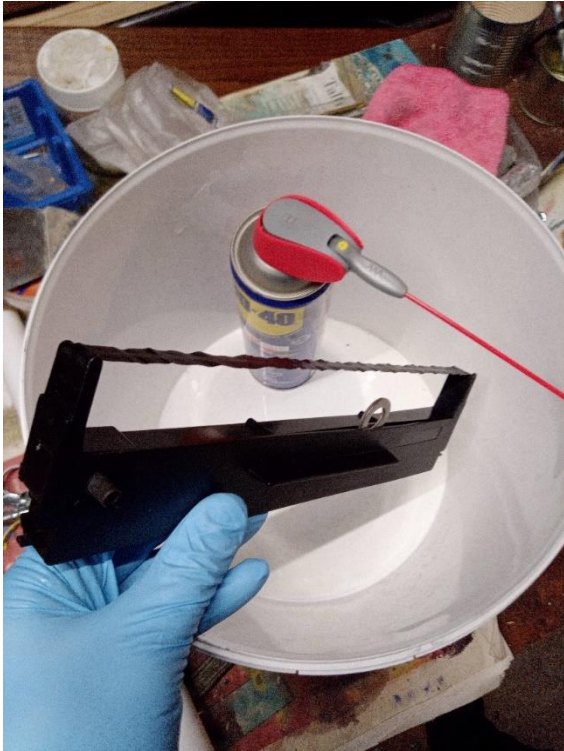


Foto 3. Tindilindi värskendamine WD-40 universaal määrdega

### 3.4 Töökorrasoleku kontroll

Epson FX-80 printeril on olemas ROM mällu salvestatud töökorrasoleku kontrolli protseduur. Selle käivitamiseks tuleb printerisse laadida paber ning printer sisse lülitada hoides all LF nuppu. [11]



Foto 4. Epson FX-80 koos laaditud paberiga

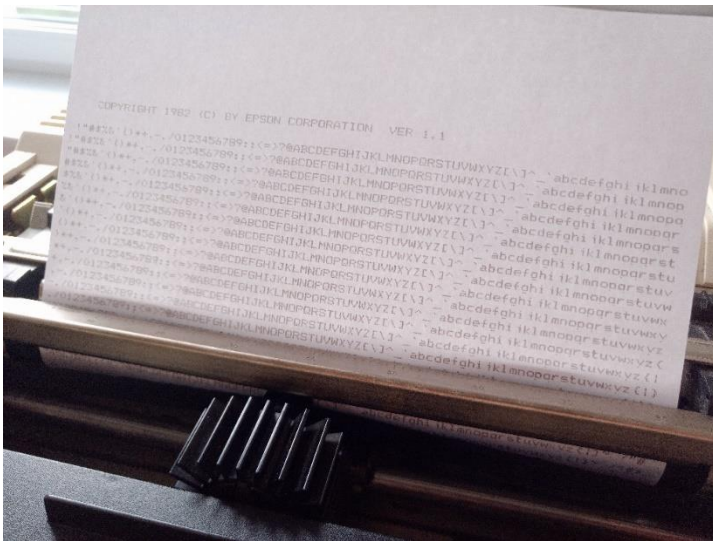


Foto 5. Epson FX-80 töökorrasoleku kontroll

Huvitava tähelepanekuna selgus töökorrasoleku kontrollist printeri tarkvara aasta 1982 ja versioon 1.1.

### 3.5 Riistvara ülesseadmine

Käesolevas projektis kasutati Arduino mikrokontrollerit ja samanimelist tarkvaraarenduskeskkonda. Arduino sai valitud peamiselt selle kasutajasõbralikkuse, kättesaadavuse ja populaarsuse tõttu. Platvormile on loodud tuhandeid erinevaid lahenduseid, mille kohta on veebist võimalik leida lugematul hulgal dokumentatsiooni ning tarkvarateeke, mis on kergesti kättesaadavad. [21]

Arduino mikroarvuti mudel (Arduino MEGA 2560) sai valitud sobiva sisend- ja väljundühenduste arvu järgi.

### 3.6 Ühendusklemmide ühendamine Arduino külge

Andmeside võimaldamiseks on tarvis ühendada kõik vajaminevad printeri klemmid Arduino sisend- ja väljundklemmidega. Andmeside protokoll kohta andsid infot nii Jaak

Pihlau 1996 ilmunud eestikeelne raamat „Printerid“ kui ka Epson FX-80 digitaalsel kujul omandatud kasutusjuhend.

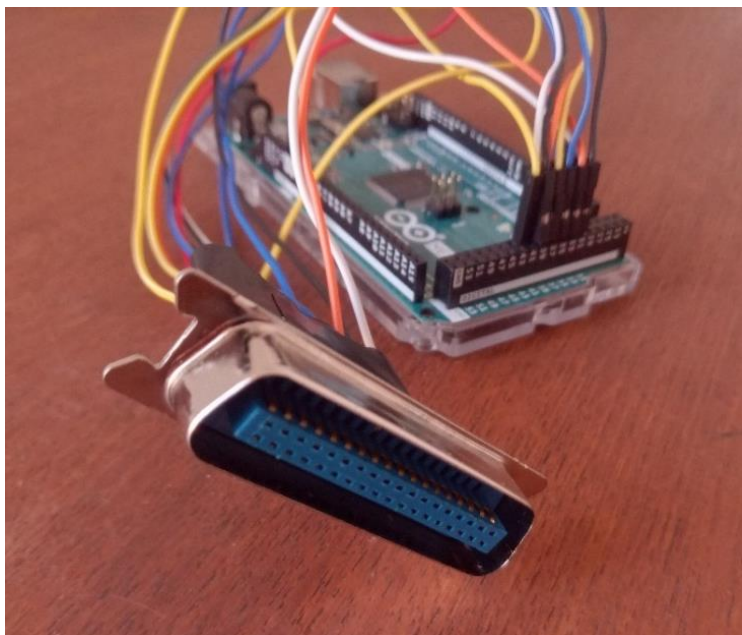


Foto 6. Arduino mikrokontroller, mille külge on ühendatud 36 klemmiga Centronics rööpliides

Peatükis 2.3 väljatoodud tabelitest 1 ja 2 on võimalik saada ülevaade vajaminevatest ühendustest. Kõige esimene ühendus on (printeripoolne) sisend tähisega  $\overline{\text{STROBE}}$  ehk andmevahetustsükli tunnusimpulss, mis annab printerile teada, et andmed, mis on andmeisiinile kirjutatud on aktiivsed ja kehtivad. Sirgjoon tähise peal tähendab seda, et signaal on inverteeritud ja selle muutumine loogiliselt madalaks nivooks tähistab seda, et arvuti on väljastanud uut infot ja printer võib info andmesiinilt välja lugeda.

Järgmised ühendusklemmid on sisendid andmetele. Kokku on printeril 8 ühendust andmetele ehk andmesiin on 8-bitine.

Klemm numbriga 10 on väljundsignaal printerist andmete vastuvõtmise valmisolekuks ning on sarnaselt andmevahetustsükli tunnusimpulsiga inverteeritud signaal. Kui signaali loogiline nivoo on madal, siis annab printer sellega teada, et andmed on vastu võetud ning ollakse valmis uute andmete vastuvõtmiseks.

Järgmine ühendus 11 märgib printeri hõivatust. Kui signaali loogiline nivoo on kõrge, siis pole printer võimeline uusi andmeid vastu võtma. Printeri kasutusjuhend [11] märgib ära,

et andmete edastamiseks võib kasutada ka ainult kas vastuvõtmise valmisoleku signaali või hõivatuse signaali. Käesoleva töö raames kasutame oma juhtprogrammis ainult hõivatuse signaali kontrollimist enne andmete saatmist, sest printer võib olla ka muudel põhjustel hõivatud, kui ainult andmete lugemine või printimistoimingu sooritamine.

Ülejäänud ühendusklemmidest saab välja tuua  $\overline{\text{INIT}}$  signaali, mis on mõeldud printeri käsustiku lähtestamiseks ning andmepuhvri tühjendamiseks. Veasignaali (klemm nr 32) me ei kasuta ning trükkimise juhtimise või teiste sõnadega printeri valimise signaali (nr 36) oleme DIP-lülititest deaktiveerinud.

### **3.7 Võimalikud edasiarendused**

Projekti võimalikud edasiarendused hõlmavad mitmekesisema funktsionaalsuse loomist. Olemasolevate teadmiste ja materjalidega on võimalik luua graafilise kujutise printimise programm, mis prindib välja graafilisele kasutajaliidesele joonistatud kujutise.

Heli mängimine sarnaselt Vimeo kasutaja MIDIDesasteriga nõuab eelkõige elektroonika-alaseid teadmiseid. Lisas 4 on väljatoodud juhised, kuidas on selline projekt ellu viidud. Lühidalt kokku võttes kontrollitakse igat printeripeas asuvat nõela eraldi signaaliga ning PWM modulatsiooni kasutades tehakse kindlaks, millistele väljundpingetele vastavad nõelte tekitatavad helikõrgused.

## **4 Tarkvara disain**

Käesoleva projekti jaoks kirjutati mikrokontrolleri tarkvara Arduino arenduskeskkonnas kasutades Arduino (eriversioon C++-ist) ning C++ programmeerimiskeeli. Graafiline kasutajaliides on loodud programmis Visual Studio kasutades *Windows Forms* töölaua rakendust ning C# programmeerimiskeelt. *Windows Forms* töölaua rakendus sai valitud peamiselt seetõttu, et antud platvorm on töö autorile tuttav, ühtib antud projekti eesmärkidega ning C# keeles on olemas jadaliideselega ühendumiseks vajaminevad teigid.

Tarkvara kirjutamisel lähtuti objektorienteeritud programmeerimise põhimõtetest, kus printerist loodi abstraktne objekt, millele loodi vajaminevad funktsioonid (nt *sendByte*).

Tarkvara on üles ehitatud paragrahvis 2.4 tehtud signaalide ja vajaminevate intervallide analüüsi tulemustest.

C++ keeles talletatakse tähemärgid nendele vastava ASCII koodiga, mistõttu sai vahepealse tähemärgi teisendamise ASCII koodiks ära jätta. Kui funktsiooni *sendByte* parameetriks lisada tähemärk, siis saadab funktsioon printerile automaatselt õige ASCII tähemärgi koodi.

## 5 Kokkuvõte

Käesoleva diplomitöö eesmärk oli luua juhtprogramm Epson FX-80 nõelmaatriksprinterit juhtimiseks Windows 10 operatsioonisüsteemis, kasutades Arduino mikrokontrollerit. Käesoleva töö tulemuseks valmis erinevates programmeerimiskeeltes kirjutatud tarkvara, mis seatud eesmärgid täitis.

Töö autor analüüsis võimalikke lahenduskäike, erinevaid mikrokontrollereid, nõelmaatriksprinterite ajalugu ja ehitust ning uuris mängulisuse osatähtsust õppeprotsessis. Analüüsi käigus sai tõestatud, et mängulisus tuleb õppeprotsessile kasuks ning kaasajal on õppetöö atraktiivseks muutmine vajalik.

Käesolev projekt võib olla esimene samm mitmele erinevale elektroonikat, tarkvaraarendust ja andmeside õpetavale projektile.

## Kasutatud materjalid

- [1] G. Barata, S. Gama, J. Jorge and D. Goncalves, “Improving participation and learning with gamification,” in *Gamification '13: Proceedings of the First International Conference on Gameful Design, Research, and Applications*, Toronto, pp. 10-17. DOI: <https://dl.acm.org/doi/10.1145/2583008.2583010> Kasutatud 15.05.2023.
- [2] *The Floppotron 3.0 - Computer Hardware Orchestra*. 2022. [Online]. Loetud aadressil: <https://www.youtube.com/watch?v=kCCXRerqaJI> Kasutatud: 01.05.2023.
- [3] *Silent's Homepage*. 2023. [Online]. Loetud aadressil: <https://silent.org.pl/home/2011/09/29/evil-floppy-drives-english-translation> Kasutatud: 01.05.2023.
- [4] *The Floppotron 3*. 2023. [Online]. Loetud aadressil: <https://silent.org.pl/home/2022/06/13/the-floppotron-3-0/> Kasutatud: 01.05.2023.
- [5] *MIDI Desaster*. 2023. [Online]. Loetud aadressil: <https://vimeo.com/mididesaster> Kasutatud: 01.05.2023.
- [6] *Device Orchestra*. 2023. [Online]. Loetud aadressil: <https://www.youtube.com/@DeviceOrchestra> Kasutatud: 01.05.2023.
- [7] J. Pihlau, *Printerid*, Tallinn: Külim, 1996.
- [8] *CENTRONICS CONNECTOR GUIDE*. [Online]. Loetud aadressil: <https://www.cablestogo.com/learning/connector-guides/centronics> Kasutatud: 01.05.2023.
- [9] *Choosing And Setting Up Optional Interfaces; Compatible Interfaces - Epson FX-286e*. [Online]. Loetud aadressil: <https://www.manualslib.com/manual/48411/Epson-Fx-286e.html?page=148> Kasutatud: 02.05.2023.



- [10] *EPSON FX SERIES PRINTER User's Manual VOLUME 1 TUTORIAL*, David A. Kater, Epson America, Inc., Torrance, California, 1984. [Online]. Loetud aadressil: [https://files.support.epson.com/pdf/fx80\\_/fx80\\_uv.pdf](https://files.support.epson.com/pdf/fx80_/fx80_uv.pdf) Kasutatud: 20.04.2023.
- [11] *FX-80 Operation Manual*, EPSON CORPORATION, Nagano, Japan, 1982. [Online]. Loetud aadressil: <https://minuszerodegrees.net/manuals/Epson%20FX-80%20Operation%20Manual.pdf> Kasutatud: 20.04.2023.
- [12] *USB->LPT Centronics printeri kaabel*. [Online]. Loetud aadressil: [https://www.oomipood.ee/product/kpo3428\\_usb\\_lpt\\_centronics\\_printeri\\_kaabel](https://www.oomipood.ee/product/kpo3428_usb_lpt_centronics_printeri_kaabel) Kasutatud: 20.04.2023.
- [13] *Epson ESC/P Reference Manual*, 1997. [Online]. Loetud aadressil: <https://files.support.epson.com/pdf/general/escp2ref.pdf> Kasutatud: 20.04.2023.
- [14] *OKI Corporate Profile*. [Online]. Loetud aadressil: <http://www.oki.com/en/profile/info/> Kasutatud: 01.05.2023.
- [15] *OKI Dot-Matrix 9Pin ESC/P Class Driver*. [Online]. Loetud aadressil: <https://www.driverscape.com/download/oki-dot-matrix-9pin-esc-p-class-driver> Kasutatud: 01.05.2023.
- [16] *What Are Advantages and Disadvantages of Arduino*. 2021. [Online]. Loetud aadressil: <https://linuxhint.com/advantages-and-disadvantages-arduino/> Kasutatud: 02.05.2023.
- [17] „*Teave Arduino Forum kohta*“. [Online]. Loetud aadressil: <https://forum.arduino.cc/about> Kasutatud: 10.05.2023.
- [18] *Raspberry Pi Pico*. [Online]. Loetud aadressil: <https://www.raspberrypi.com/products/raspberry-pi-pico/> Kasutatud: 10.05.2023.
- [19] *Arduino Mega 2560 Rev3*. [Online]. Loetud aadressil: <https://store.arduino.cc/products/arduino-mega-2560-rev3> Kasutatud: 01.05.2023.
- [20] *Reinking dot matrix printer ribbons because it's fun, okay*. 2019. [Online]. Loetud aadressil: <https://hackaday.com/2019/04/03/reinking-dot-matrix-printer-ribbons-because-its-fun-okay/> Kasutatud: 20.04.2023.

[21] K. Raid, R. Sell, *Arduino projektid alustajale*, Tallinn: Robolabor.ee kirjastus, 2017.

# Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>

Mina, Robert Arus

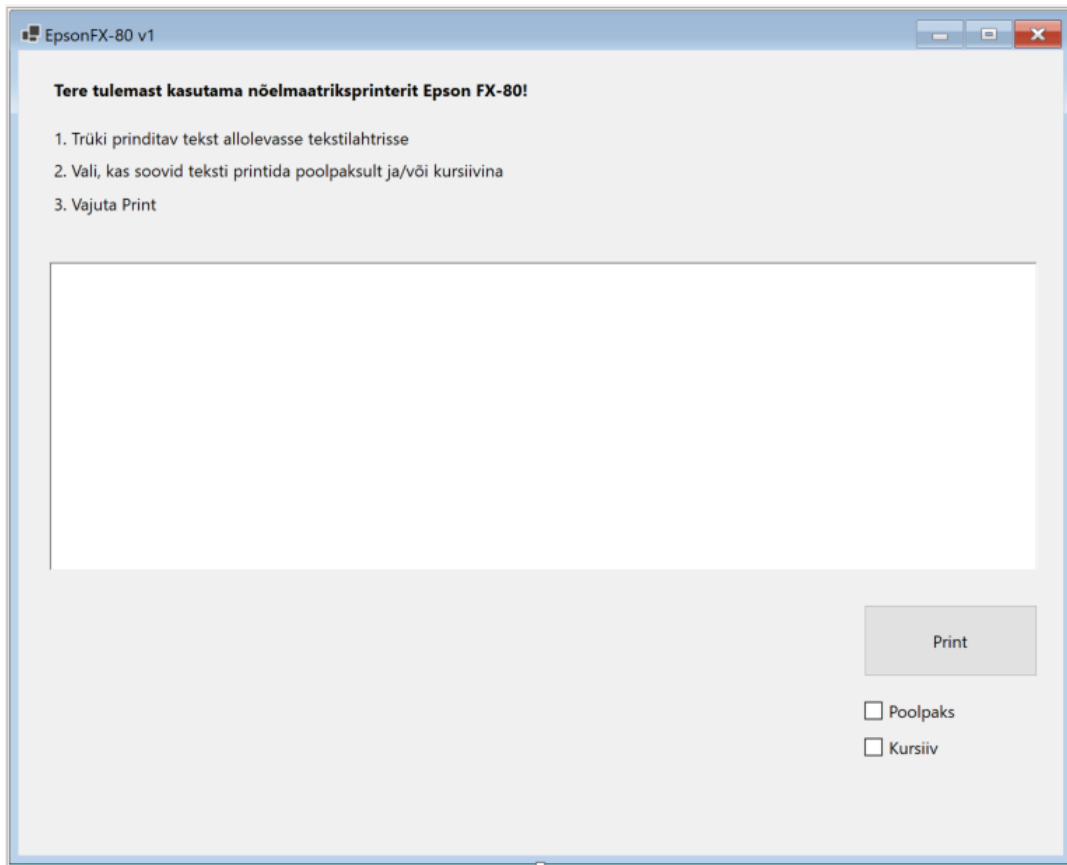
1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Nõelmaatriksprinterit juhtprogrammi loomine, kasutades Arduino mikrokontrollerit“, mille juhendaja on Priit Ruberg.
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli 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 käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

06.05.2023

---

<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## Lisa 2 – Graafiline kasutajaliides ja selle kood



Joonis 4. Graafiline kasutajaliides

```
1 namespace Epson_FX_80_Driver_v1
2 {
3     partial class EpsonFX80
4     {
5         /// <summary>
6         /// Required designer variable.
7         /// </summary>
8         private System.ComponentModel.IContainer components = null;
9
10        /// <summary>
11        /// Clean up any resources being used.
12        /// </summary>
13        /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
14        protected override void Dispose(bool disposing)
15        {
16            if (disposing && (components != null))
17            {
18                components.Dispose();
19            }
20            base.Dispose(disposing);
21        }
22
23        Windows Form Designer generated code
24
25        private RichTextBox outputTextBox;
26        private Button printButton;
27        private CheckBox boldCheckBox;
28        private CheckBox italicCheckBox;
29        private Label label1;
30        private Label label2;
31        private Label label3;
32        private Label label4;
33    }
34 }
```

Joonis 5. Graafilise kasutajaliidese kujunduse kood

```

1  using System.IO.Ports;
2
3  namespace Epson_FX_80_Driver_v1
4  {
5      3 references
6      public partial class EpsonFX80 : Form
7      {
8
9          1 reference
10         public EpsonFX80()
11         {
12             InitializeComponent();
13             init();
14         }
15
16         1 reference
17         private void printButton_click(object sender, EventArgs e)
18         {
19             String bold_checked = "";
20             String italic_checked = "";
21
22             if (boldCheckBox.Checked)
23             {
24                 bold_checked = ".B";
25             }
26
27             if (italicCheckBox.Checked)
28             {
29                 italic_checked = ".I";
30             }
31
32             serialPort.WriteLine(bold_checked + italic_checked + " " + outputTextBox.Text);
33             serialPort.DiscardOutBuffer();
34         }
35
36         1 reference
37         private void init()
38         {
39             try
40             {
41                 serialPort = new SerialPort();
42                 serialPort.BaudRate = 9600;
43                 serialPort.PortName = "COM5";
44                 serialPort.Open();
45                 MessageBox.Show("Serial port " + serialPort.PortName + " connected successfully!");
46             }
47             catch (Exception)
48             {
49                 MessageBox.Show("Something went wrong!");
50             }
51         }
52     }

```

Joonis 6. Windows Forms programmi kood

## Lisa 3 – Arduino IDE-s kirjutatud programmi kood

```
1 #include "epson_fx80_driver.h"
2 #include "helpers.h"
3
4 //Declare the printer object
5 EpsonFX80 printer;
6
7 //Set up the Arduino pins
8 const byte arduino_dataBus[8] = {31,32,33,38,39,34,35,36}; // Data bus pins (D0-D7)
9 const byte arduino_strobePin = 30; // Strobe pin (pin 10 on Arduino Uno)
10 const byte arduino_ackPin = 37; // Acknowledge Pin
11 const byte arduino_busyPin = 40; // Busy Pin
12 const byte arduino_initPin = 41; // Initialize Pin
13
14 void setup() {
15
16     //Set up serial communications
17     Serial.begin(9600);
18
19     //Initialize the printer, mapping the pins, etc
20     printer.init(arduino_strobePin, arduino_dataBus, arduino_ackPin, arduino_busyPin);
21
22 }
23
24 void loop() {
25
26     //Either just text or .B.I Text
27     String data = getSerialData(); //Get the data from the serial input
28
29     if (data.length() != 0) {
30         getAndActivateOptions(data);
31         printer.printOut(cleanedInput(data)); //The cleanedInput function removes the options from the input
32     }
33
34     delay(50);
35 }
36
37 void getAndActivateOptions(String data) {
38
39     // Get the flags from the data string
40     String options = data.substring(0, data.indexOf(' '));
41
42     // Check if ".B" is present in the first word
43     if (options.indexOf(".B") != -1) {
44         printer.sendByte(27); // Activate Bold text
45         printer.sendByte('G');
46     } else {
47         // If there is no option then need to restore to defaults
48         printer.sendByte(27); // De-activate Bold text
49         printer.sendByte('H');
50     }
51
52     // Check if ".I" is present in the first word
53     if (options.indexOf(".I") != -1) {
54         printer.sendByte(27); // Activate Italic
55         printer.sendByte('4');
56     } else {
57         // If there is no italic
58         printer.sendByte(27); // De-activate Italic
59         printer.sendByte('5');
60     }
61 }
```

Joonis 7. Faili „epson\_fx80\_driver.ino“ kood

```

1  #ifndef EPSON_FX80_DRIVER_H
2  #define EPSON_FX80_DRIVER_H
3  #include <Arduino.h>
4  #include <string.h>
5  class EpsonFX80
6  {
7  public:
8      //Parameters
9      //Output
10     byte strobePin;
11     byte dataPins[8];
12     //Input
13     byte ackPin;
14     byte busyPin;
15
16     //Functions
17     EpsonFX80(); //Constructor
18     void init(const byte, const byte*, const byte, const byte);
19     void printOut(String);
20     void sendByte(byte);
21     void resetPrinter();
22 };
23 #endif

```

Joonis 8. Faili "epson\_fx80\_driver.h" kood

```

1  #include "epson_fx80_driver.h"
2
3  EpsonFX80::EpsonFX80 ()
4  {
5      //Empty
6  }
7
8  void EpsonFX80::init(const byte arduino_strobePin, const byte* arduino_dataBus, const byte arduino_ackPin, const byte arduino_busyPin) {
9
10     //Map the Arduino IO to the Epson printer class IO (data bus will be done in the for-loop later)
11     strobePin = arduino_strobePin;
12     ackPin = arduino_ackPin;
13     busyPin = arduino_busyPin;
14
15     //Set all pin modes
16
17     //Map the data bus IO and set pin modes
18     for(int i = 0; i<8; ++i)
19     {
20         dataPins[i] = arduino_dataBus[i];
21         pinMode(dataPins[i], OUTPUT);
22         digitalWrite(dataPins[i], LOW);
23     }
24
25     pinMode(strobePin, OUTPUT);
26
27     pinMode(ackPin, INPUT);
28     pinMode(busyPin, INPUT);
29
30     //Set Default Values
31     digitalWrite(strobePin, HIGH);
32
33     // Reset the printer
34     resetPrinter();
35 }
36
37
38 void EpsonFX80::printOut(String data)
39 {
40     for(int i = 0; i<data.length(); ++i)
41     {
42         sendByte(data.c_str()[i]);
43     }
44
45     sendByte(13);
46     sendByte(10);
47 }
48
49 void EpsonFX80::sendByte(byte data) {
50
51     // Send it to Serial Monitor for troubleshooting
52     uint8_t k = data;
53     Serial.println(k);
54
55     // Don't send data until printer is ready for it
56     while(digitalRead(busyPin)==HIGH)//Printer is busy
57     {
58         Serial.println("Waiting for Printer to be ready");
59     }
60
61     // Send a byte of data to the printer
62     for (byte i = 0; i < sizeof(dataPins); i++) {
63         digitalWrite(dataPins[i], bitRead(data, i));
64     }
65
66     //Wait at least 50us (extended deliberately) to provide time for the data to be ready
67     delayMicroseconds(50);
68
69     //Set the strobe pin high for at least 500us to indicate that data is ready
70     digitalWrite(strobePin, HIGH);
71     delayMicroseconds(50);
72
73     //Set the strobe pin low to send the data to the printer
74     digitalWrite(strobePin, LOW);
75 }
76
77 void EpsonFX80::resetPrinter() {
78     //Set the data bus pins to low
79     for (byte i = 0; i < sizeof(dataPins); i++) {
80         digitalWrite(dataPins[i], LOW);
81     }
82
83     // Set the strobe pin low for at least 500us to reset the printer
84     digitalWrite(strobePin, LOW);
85     delayMicroseconds(500);
86     digitalWrite(strobePin, HIGH);
87 }

```

Joonis 9. Faili "epson\_fx80\_driver.h" kood



```

1  #ifndef HELPERS_H
2  #define HELPERS_H
3  #include <Arduino.h>
4
5  String cleanedInput(String);
6  String getSerialData();
7
8  #endif

```

Joonis 10. Faili "helpers.h" kood

```

1  #include "helpers.h"
2
3  String getSerialData() {
4  |   String data = "";
5  |   while (Serial.available() > 0) {
6  |       Serial.println("Reading data");
7  |       char c = Serial.read();
8  |       data += c;
9  |   }
10 |
11 |
12 |   if (data != "") { // If we received some data
13 |       Serial.print("Received data: ");
14 |       Serial.println(data); // Print the received data to the Serial Monitor for logging
15 |   }
16 |   return data;
17 | }
18
19 String cleanedInput(String data) {
20 |   bool foundSpace = false;
21 |   String result = "";
22 |   for (int i = 0; i < data.length(); i++) {
23 |       char c = data.charAt(i);
24 |       if (c == ' ' && !foundSpace) {
25 |           foundSpace = true;
26 |       } else if (foundSpace) {
27 |           result += c;
28 |       }
29 |   }
30 |   return result;
31 | }

```

Joonis 11. Faili "helpers.cpp" kood

## Lisa 4 – Kirjavahetus Vimeo kasutaja MIDIDesaster-iga

**MIDIDesaster** 1 month ago

Hi Robert!

This project is pretty old and I don't have any documentation. You probably already watched this video and read the description, this contains some explanations:

**[vimeo.com/57960146](https://vimeo.com/57960146)**

The build process was like this:

- \* Disassemble the printer and identify the pin drivers. This may be a few ICs or discrete transistors.
- \* Identify inputs to the pin drivers. These should be connected to the main controller.
- \* Let the printer print (test page feature) and measure voltage level, pulse width and polarity of the driving signals. Then you have an idea of the duty cycle the printer uses. Later you should not exceed that for a longer time to avoid overheating.
- \* Cut the trace between the controller and one of the pin drivers and connect a PWM generator instead. This may be a signal generator or some micro controller. Play around with frequencies and pulse widths to identify the limits for volume and frequency. Very short pulses don't fully eject the pin so it does not hit the paper (quieter sound). When the pulse width gets longer, the paper will be hit and the sound gets immediately louder. If frequency and duty cycle is high, the pin may get stuck at the paper because the retracting spring cannot push the pin back fast enough. This causes a jam of the ink ribbon when the print head moves around.
- \* Keep the signal active only for short pulses. Long active time kills the coil in the print head (this happened to me). Make sure to put pull-up or pull-down (depending on signal polarity) resistors in place to get an inactive driving signal when the PWM generator is not connected or during reset or when PWM IO of the controller is not configured yet.
- \* Now you know how to generate one note. For polyphony connect all the other pin drivers as well.
- \* Back then there was no microcontroller which can generate more than 20 PWM signals with independent frequency. If you look at datasheets you may find lots of PWM outputs but they usually share the same timer units, so watch out. Software based PWM is usually too slow and not precise enough. I used an FPGA and synthesized lots of PWM units and connected that via SPI to an Atmega8. Today I would probably put a soft core (e.g. RISC-V) into the FPGA and have all in one chip. As an alternative maybe the Raspberry Pi Pico controller RP2040 can generate independent PWMs with its programmable IO feature but I didn't look into that. Or there are controllers which can use DMA to push data to an IO port, I think the ESP32 can do this but this would probably be much harder to implement.
- \* Later I connected the two stepper motors (print head and paper feed) to the Atmega to fake print head movement (the original movement is much too fast). The light barrier to detect carriage return was also connected to know when to stop the movement.
- \* Then deep dive into the MIDI protocol to play notes from a keyboard or MIDI files.
- \* In addition to the "play note" commands I also implemented pitch bending to allow stuff like this:

**[vimeo.com/56412514](https://vimeo.com/56412514)**

It was a fun (and long) project, touching many interesting topics: reverse engineering, micro controller and FPGA programming, MIDI protocol, relation of musical notes to frequencies and lots of soldering. All calculations for notes, frequencies and resulting PWM parameters are done in fixed point on an 8 bit controller.

Have fun!