



Šolski center Celje
Srednja šola za kemijo, elektrotehniko in računalništvo

NAPRAVA ZA AVTOMATSKO REZANJE IN LUPLJENJE VODNIKOV

(raziskovalna naloga)

Mentor:
Gregor Kramer, univ. dipl. inž. el.

Avtorja:
Leon Kovše, E4a
Aljaž Perc, E4a

Celje, marec 2020

KAZALO VSEBINE

1 POVZETEK	1
1.1 KLJUČNE BESEDE	1
2 UVOD	2
2.1 PREDSTAVITEV RAZISKOVALNEGA PROBLEMA.....	2
2.2 HIPOTEZE	2
2.3 OPIS RAZISKOVALNIH METOD	3
3 OSREDNJI DEL NALOGE	4
3.1 PREDSTAVITEV REZULTATOV RAZISKOVANJA	4
3.1.1 Potek dela	4
3.1.2 Opis komponent.....	5
3.1.2.1 <i>Arduino Mega</i>	5
3.1.2.2 <i>Nextion</i>	6
3.1.2.3 <i>Koračni motor</i>	6
3.1.2.4 <i>Gonilniki</i>	7
3.1.2.5 <i>Servomotor</i>	7
3.1.2.6 <i>Napajalnik</i>	8
3.1.2.7 <i>Ventilator</i>	9
3.1.2.8 <i>Klešče</i>	9
3.1.2.9 <i>Rele</i>	9
3.1.3 Potek izdelave ohišja	10
3.1.3.1 <i>Laserski izrez ohišja</i>	10
3.1.3.2 <i>Upogibanje pločevine</i>	11
3.1.3.3 <i>Struženje</i>	14
3.1.3.4 <i>Vrtanje, grezenje in vrezovanje navojev</i>	15
3.1.4 Vizualni prikaz programa.....	15

3.1.5	Vezalni načrt	19
3.1.6	Programiranje.....	21
3.1.6.1	Programiranje zaslona	21
3.1.6.2	Programiranje mikrokrmilnika	26
3.1.6.2.1	Programiranje koračnih motorjev	26
3.1.6.2.2	Programiranje Nextiona	27
3.1.6.2.3	Programiranje servomotorja	29
3.1.6.2.4	Program za ventilator.....	29
3.1.6.2.5	Združen program	30
3.1.7	Testiranje	35
3.2	RAZPRAVA	36
4	ZAKLJUČEK	37
5	VIRI IN LITERATURA	38
7	ZAHVALA	39

KAZALO SLIK

Slika 1:	Sušilci plastičnih granulatov	2
Slika 2:	Arduino Mega 2560.....	5
Slika 3:	Zaslon na dotik Nextion	6
Slika 4:	Koračni motor NEMA 17	7
Slika 5:	Gonilnik A4988.....	7
Slika 6:	Servomotor	8
Slika 7:	Napajalnik	8
Slika 8:	Ventilator.....	9
Slika 9:	Klešče	9
Slika 10:	Rele	10
Slika 11:	Laser za izrez pločevine	10
Slika 12:	Načrt ohišja za laserski izrez.....	11
Slika 13:	Stroj za upogibanje pločevine	11
Slika 14:	3D prikaz ohišja	12

Slika 15: 3D prikaz nosilca.....	12
Slika 16: 3D prikaz pozicije nosilca.....	13
Slika 17: 3D prikaz pozicije nosilca.....	13
Slika 18: Stružnica	14
Slika 19: Kolo s pušo	14
Slika 20: Diagram poteka.....	18
Slika 21: Vezalna shema	19
Slika 22: Nextion - konektor.....	20
Slika 23: Nextion Editor	21
Slika 24: Nextion - prva stran.....	22
Slika 25: Nextion - druga stran	22
Slika 26: Nextion - tretja stran.....	23
Slika 27: Nextion - četrta stran.....	23
Slika 28: Nextion - peta stran.....	24
Slika 29: Nextion - šesta stran	24
Slika 30: Nextion - sedma stran.....	25
Slika 31: Nextion - osma stran.....	25
Slika 32: Program - koračni motor	26
Slika 33: Program - Nextion 1	27
Slika 34: Program - Nextion 2.....	28
Slika 35: Program - Servomotor	29
Slika 36: Program - ventilator	29
Slika 37: Program - združen	34
Slika 38: Testiranje	35

KAZALO GRAFOV IN TABEL

Tabela 1: Ovrednotenje hipotez.....	37
-------------------------------------	----

1 POVZETEK

Za raziskovalno nalogo sva se odločila izdelati napravo, ki bo avtomatsko rezala in lupila vodnike. Podjetje KO-Lovec, d. o. o., nam je v svoji proizvodnji pokazalo, kako poteka priprava vodnikov pri njih. Najina naloga je bila izdelati napravo, ki bi sama rezala in lupila vodnike s prerezom $0,5 \text{ mm}^2$ in bi s tem razbremenila delo zaposlenega. To bi pospešilo proizvodnjo in povečalo produktivnost podjetja.

Kot pri vsakem delu se je bilo najprej potrebno organizirati in narediti načrt. Sledilo je naročanje in nabava komponent, nato namestitev le-teh v ohišje. Ko je bil ta del naloge zaključen, sva začela s programiranjem mikrokontrolnika in zaslona ter nato s testiranjem in odpravljanjem manjših težav.

1.1 KLJUČNE BESEDE

Naprava za avtomatsko rezanje in lupljenje vodnikov, Arduino Mega 2560, Nextion, Nema 17 – koračni motor.

2 UVOD

2.1 PREDSTAVITEV RAZISKOVALNEGA PROBLEMA

Med izdelavo sušilcev plastičnih granulatov (Slika 1) je potrebno pripraviti vodnike glede na dolžino, prerez, uporabnost oziroma barvo izolacije. S ciljem, da delavec ne bi več ur na teden porabil za rezanje in pripravo vodnikov, sva se odločila, da bova raziskala, kako bi ta problem odpravila. V raziskovalni nalogi sva se lotila projektiranja naprave za avtomatsko rezanje in lupljenje vodnikov prereza $0,5 \text{ mm}^2$. Raziskovalna naloga zajema zbiranje idej, načrtovanje, konstruiranje in izdelavo naprave. Da bo naprava delovala, tako kot sva si zamislila, sva elemente, preden sva jih kupila, preučila ter primerjala s kakšnim drugim elementom na trgu, ki bi bil prav tako ustrezen. Z nabavo kakovostnih elementov sva zagotovila večjo natančnost delovanja naprave, ki bi pospešila proizvodnjo in zagotovila skoraj nemoteno delo delavcev.



Slika 1: Sušilci plastičnih granulatov

2.2 HIPOTEZE

V raziskovalni nalogi sva postavila naslednje hipoteze:

1. Naprava izvaja avtomatsko rezanje in lupljenje vodnikov prereza $0,5 \text{ mm}^2$.
2. Proces poteka dvakrat hitreje kot ročno delo.
3. Posluževanje naprave se izvaja preko zaslona na dotik.
4. Naprava je v celoti mobilna.
5. Stroški naprave se povrnejo v enem letu.

2.3 OPIS RAZISKOVALNIH METOD

Opredelitev problema in možnih rešitev

V prvem delu raziskovalne naloge sva idejo raziskala, nato jo predstavila podjetju KO-Lovec, ki nama je svetoval, kako skonstruirati in izdelati napravo za avtomatsko rezanje in lupljenje vodnikov, idejo potrdilo ter zagotovilo finančna sredstva.

Raziskava trga

V drugem delu raziskovalne naloge sva se lotila podrobnega preučevanja komponent ter načrtovanja naprave. Začela sva z naročanjem vseh sestavnih komponent. Ko sva vse potrebne komponente dobila, sva se osredotočila na konstruiranje naprave.

Sestava in testiranje

V tretjem delu sva se lotila izdelave naprave. Najprej sva izdelala jekleno konstrukcijo, na katero sva nato zmontirala vse sestavne komponente in jih med sabo ožičila. Sledilo je programiranje mikrokrmilnika in zaslona, preizkus delovanja in kasnejša končna sestava.

3 OSREDNJI DEL NALOGE

3.1 PREDSTAVITEV REZULTATOV RAZISKOVANJA

3.1.1 Potek dela

Preden sva začela z izdelavo naprave, sva se morala odločiti, katere komponente bova uporabila in kako bo potekala komunikacija med njimi. Za krmiljenje naprave sva se odločila uporabiti mikrokrmilnik Arduino Mega 2560, saj smo imeli že v šoli veliko opravka z njim. Na spletu sva zasledila podobne naprave, med katerimi je večina uporabljala zaslon na dotik Nextion. Zaradi večje preglednosti, boljše kvalitete in lažje uporabe sva se odločila za zaslon Nextion s diagonalo 17.8 centimetrov. Za prenos vodnika sva s pomočjo eksperimentiranja prišla do ugotovitve, da bi bilo najbolje uporabiti kar štiri kolesa, ki bi s pomočjo koračnih motorjev »vlekla« vodnik. Za koračne motorje sva se odločila predvsem zaradi enostavnosti, natančnosti, vzdržljivosti in predvsem nizke cene glede na ostale pogonske krmilne sisteme. Nadzor korakov in smeri koračnih motorjev sva izvedla z uporabo gonilnikov A4988, ki služi tudi kot zaščita motorjev, saj ima izklopno vezje, ki pri nepravilni količini toka, napetosti ali v primeru pregrevanja izklopi motorje. Iskala sva primerno rešitev, kako bova rezala in lupila vodnike ter s čim. Sprva sva to hotela izvesti s pomočjo pnevmatike, vendar sva nato na spletu našla rezalni sistem, ki je sestavljen iz klešč ter servomotorja in je cenovno bolj ugoden, tišji in lažje izvedljiv. Pri izboru pravih klešč sva morala biti pozorna predvsem na obliko rezil, a jih nisva našla v trgovinah, zato sva prebrskala spletne trgovine ter izbrala tiste, ki so ponujale najboljšo kakovost klešč za najnižjo ceno. Da bi motorika pri rezalnem delu naloge potekala brez večjih zapletov, sva se odločila nabaviti kar servomotor z močjo 35 kilogramov, ki je izjemno močen in vzdržljiv, prav tako pa porabi malo energije. Da se komponente ne bodo pregrevale, bova v napravo dodala tudi ventilator. Postavitev ventilatorja je odvisna od postavitve ostalih komponent, še posebej tistih, ki se najbolj segrevajo. Ko sva dobila vse potrebne komponente, sva naredila približno postavitev, nato pa sva s pomočjo zaposlenih v podjetju KO-Lovec naredila kovinsko strukturo naprave.

3.1.2 Opis komponent

3.1.2.1 Arduino Mega

To je mikrokontroler z vgrajenim čipom ATmega 2560, ki deluje s hitrostjo 16 MHz. Ima 54 digitalnih vhodno-izhodnih priključkov (I/O), kar pomeni, da lahko zaznava ali oddajo logično enko ali ničlo. Od teh 54 priključkov jih 15 podpira PWM, 16 vhodov je analognih, od katerih jih lahko 9 služi tudi kot izhod. Deluje na napetosti 5 V, kot vir napetosti nudi napetost 5 V in tudi 3.3 V. Na posameznem priključku je maksimalni izhodni tok 40 mA, celotni izhodni tok praviloma ne sme preseči 200 mA. Vgrajena je tudi tipka RESET, dioda za zaščito pred nepravilno polariteto, ter varovalka, ki pri toku 500 mA izklopi ploščico in jo ponovno vklopi šele, ko se vrednost zmanjša pod to mejo. Mikrokontroler lahko napajamo na več načinov. Lahko ga napajamo preko 2.1 mm vtičnice, ki zahteva napetost med 6 V in 20 V, vendar je bolj priporočeno območje med 7 V in 12 V. Naslednji način napajanja je s 5 V preko vhoda USB, kjer je omejitev maksimalnega vhodnega toka 250–500 mA in je odvisen od računalnika. Zadnji način je preko 5 V priključka na plošči, ki ga običajno uporabimo za napajanje vezja, vendar ga lahko uporabimo tudi za napajanje plošče. S tem načinom obidemo zaščitno diodo kot tudi varovalko, zato je posebej potrebna previdnost. Pri izdelavi naprave sva uporabila zadnji način napajanja, zato sva potrebovala še pretvornik napetosti, ki vhodnih 230 V iz omrežja pretvori v 5 V, ki služijo napajanju Arduina. Mikrokontroler v najini nalogi komunicira z zaslonom na dotik Nextion ter z njim upravlja.



Slika 2: Arduino Mega 2560

3.1.2.2 Nextion

Nextion je zaslon na dotik, kar pomeni, da gre za računalniško vhodno-izhodno enoto, ki s površino, namenjeno grafičnemu prikazovanju podatkov, zaznava tudi dotik uporabnikovega prsta ali ustreznega predmeta (npr. pisala), s katerim lahko nastavljam določene funkcije. To omogočimo v programski opremi Nextion Editor, v kateri s komponentami povleci-in-spusti (grafika, besedilo, gumb, drsnik itd.) in z ASCII besedilnimi navodili za kodiranje zapišemo, kako se bo komponenta odzvala ob določenem ukazu. Uporabila sva 7.0 inch (17.78 cm) kapacitivni zaslon z ohišjem, s pomnilnikom 32MB Flash, 1024 bajtov EEROM, 8192 bajtov RAM in GPIO.



Slika 3: Zaslon na dotik Nextion

3.1.2.3 Koračni motor

Gre za električne motorje brez krtačk. Te so pogost vir okvare pri električnih motorjih, zato imajo koračni motorji zagotovljeno bistveno daljše delovanje kot motorji s krtačkami. Zaradi tega je uporaba koračnih motorjev primerna tam, kjer se zahteva nepretrgano delovanje in delovanje brez okvar. Je elektromehanski del, ki pretvori električni signal v mehansko rotacijo osi. Vsak pomik osi za točno določen kot se izvede ob ustreznem električnem pulzu; npr. ko koračnemu motorju dovedemo napetost, se obrne za en korak, ki predstavlja 1.8° , kar pomeni, da potrebuje motor za en cel obrat 200 korakov ($200 \times 1.8^\circ = 360^\circ$).

Poznamo 2 tipa koračnih motorjev: unipolarne in bipolarne. Pri unipolarnih koračnih motorjih so vsa navitja povezana v eni točki in so enostavnejši za krmiljenje. Bipolarni nimajo navitja povezanega med sabo v eni točki, zato imajo boljše razmerje med velikostjo in navorom kot unipolarni.

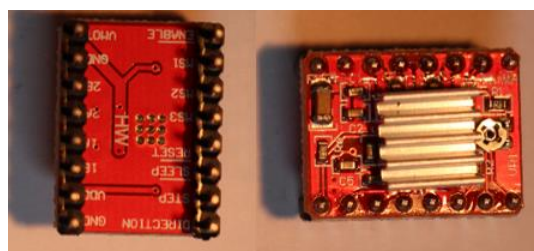
Odločila sva se za NEMA 17 bipolarne 2.0 A motorje, saj bova dolžino vodnikov merila s pomočjo mehanske rotacije osi. So tudi enostavni za uporabo, natančni in zanesljivi.



Slika 4: Koračni motor NEMA 17

3.1.2.4 Gonilniki

Gonilniki koračnih motorjev so posebej zasnovani za pogon koračnih motorjev in so sposobni natančno nadzirati položaj osi tudi ob neprekinjenem vrtenju. Gonilniki nudijo nastavljivo krmiljenje toka in večstopenjsko ločljivost, poleg tega pa imajo vgrajene prevajalnike, ki omogočajo krmiljenje koračnega motorja s preprostimi koračnimi in smernimi vhodi. Ti moduli lahko ponujajo vhode za neposredno sprožitev vsakega koraka. Za generiranje teh signalov je običajno potreben zunanji mikrokrmilnik. V najini nalogi bova uporabljala A4988 gonilnike, ki so nama v poteku testiranja povzročali veliko preglavic, saj so se hitro pregrevali.



Slika 5: Gonilnik A4988

3.1.2.5 Servomotor

Gre za motor, ki lahko rotira v obsegu 180° (90° v eno in 90° v nasprotno smer) in se krmilijo z impulzi, katerih širina se giblje od 1 do 2 ms. Impulzi si sledijo v določenih časovnih presledkih, ki se lahko gibljejo med 10 in 30 ms. Kot zasuka je odvisen od širine impulza, kar pomeni, če motorju pošiljamo impulze konstantne širine, le-ta

doseže določen zasuk, ki ga do določene obremenitve tudi ohranja. S servomotorjem bova stiskala ter razpirala klešče, pri tem pa bova s pravilnim časom dobila tudi kot, ki bo potreben, da se bodo klešče stisnile do te mere, da bo možno z vodnika sneti izolacijo.



Slika 6: Servomotor

3.1.2.6 Napajalnik

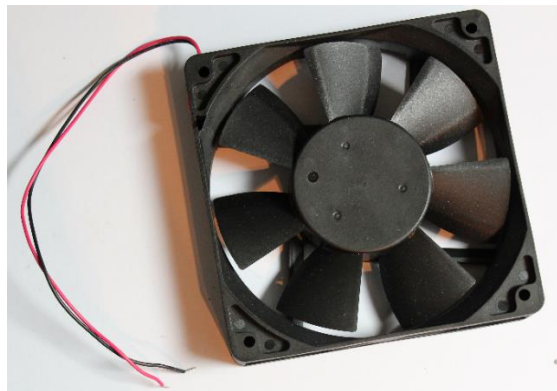
Napajalnik je naprava, namenjena pretvorbi električne omrežne napetosti 230 V v manjše električne napetosti, ki jih različne komponente potrebujejo za svoje delovanje. V napravi bova imela napajalnik, ki bo pretvoril vhodnih 230 V v izhodnih 5 V in bo z njimi napajal mikrokrmilnik, koračne motorje in servomotor.



Slika 7: Napajalnik

3.1.2.7 Ventilator

Ventilator je naprava, ki s potiskanjem zraka ali drugih plinov ustvarja zračni tok oziroma umetni veter. Lahko deluje kot nizkotlačna črpalka, kot centrifugalna črpalka, batni ali izrivalni ventilator (npr. meh), najbolj razširjen in primeren za hlajenje komponent v napravi pa je ventilator z vetrnico. Uporabila sva ventilator z vetrnico, ki ga poganja 12 V. Gre za ventilator za hlajenje računalniške opreme, za katere je značilno, da uporabljajo izključno motorje na enosmerni tok brez ščetk, saj proizvajajo manj elektromagnetnih motenj.



Slika 8: Ventilator

3.1.2.8 Klešče

Klešče so ročno orodje, sestavljene iz čeljusti, po katerih jih med sabo ločimo (ploščate, koničaste, polokrogle itd.), ter ročaja, ki ga z objemom dlani stiskamo. Za rezanje in lupljenje vodnikov sva uporabila klešče, ki imajo čeljusti v obliki črke V (Southwire tools & equipment).

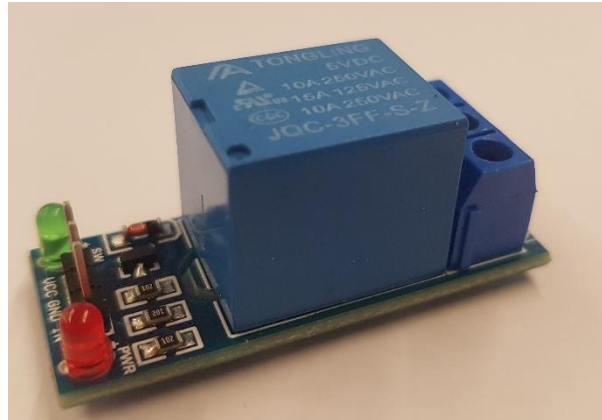


Slika 9: Klešče

3.1.2.9 Rele

Je elektromagnetno stikalo, ki ga vklapljamo s krmilno napetostjo, krmilimo pa s tokom, ki povzroči magnetni pretok. Ko je magnetni pretok dovolj velik, magnetna sila pritegne kvoto k jedru, s tem se sklenejo kontaktna peresa. Ko releju odvzamemo napetost, mu

prav tako odvzamemo tok in s tem magnetni pretok, kar povzroči, da magnetna sila popusti in se kontakti vrnejo v prvoten položaj.



Slika 10: Rele

3.1.3 Potek izdelave ohišja

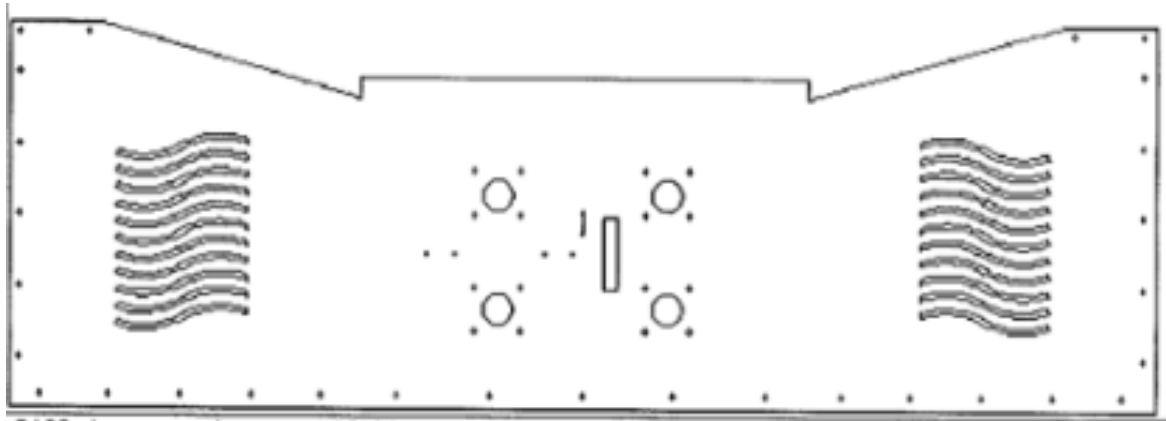
3.1.3.1 Laserski izrez ohišja

Laserski razrez pločevine in drugih kovin je napreden tehnološki postopek, ki dandanes poteka s pomočjo sodobnih strojev za razrez in omogoča visoko kakovost izdelkov, hkrati pa izjemno poenostavi nadaljnji proizvodni postopek. Bistvena prednost, ki jo prinaša rezanje z laserjem, je kakovostna, natančna, hitra in cenovno ugodna izvedba.

V podjetju KO-Lovec so nama na laserju TruLaser 3030 (slika 11) izrezali ohišje naprave.



Slika 11: Laser za izrez pločevine



Slika 12: Načrt ohišja za laserski izrez

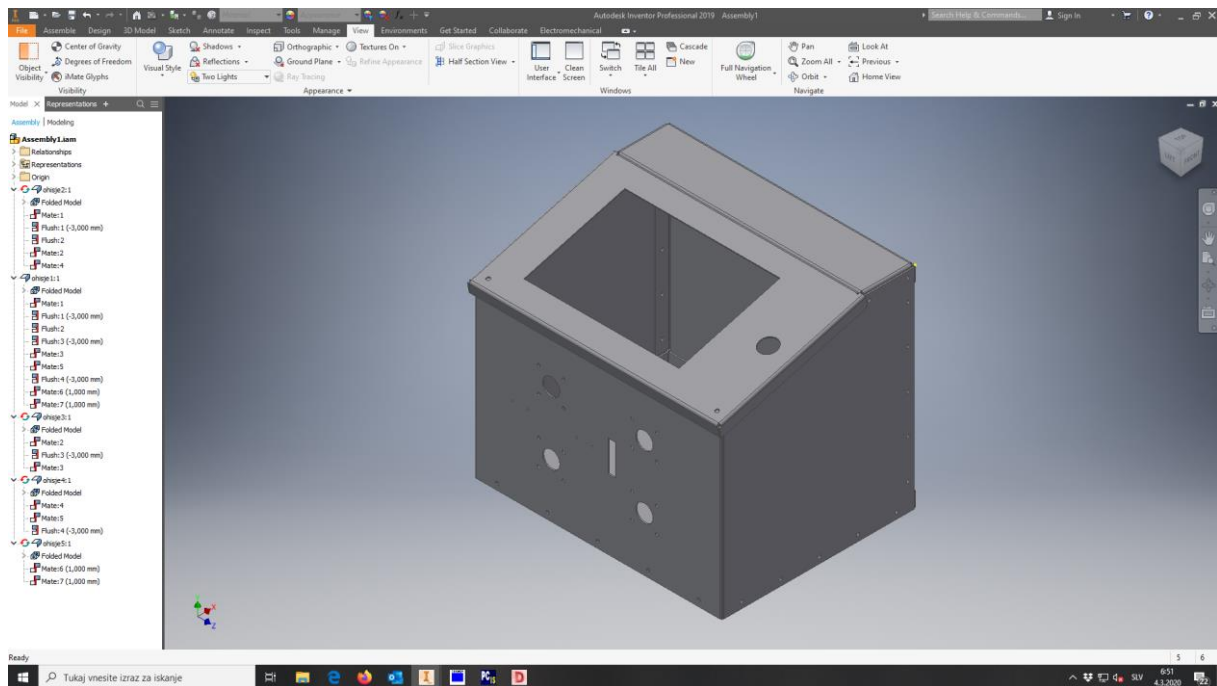
3.1.3.2 Upogibanje pločevine

Upogibanje ali krivljenje pločevine je eden najbolj pogostih in pomembnih postopkov v predelovalni industriji in je uvrščena med najbolj razširjene postopke predelave oblike pločevine, profilov in cevi. Upogibanje pločevine je plastična deformacija dela nad osjo, ki ustvarja geometrijske spremembe obdelovanega materiala. Pločevina se mora zaradi elastične izravnave upogniti za večji kot od zahtevanega, v nasprotnem primeru je krivina po razbremenitvi manjša. Podobno kot drugi procesi obdelovanja kovine ukrivljanje spremeni obliko obdelovanega kosa, medtem ko volumen materiala ostane enak. V nekaterih primerih lahko ukrivljanje povzroči manjšo spremembo debeline pločevine. Z upogibanjem dobimo želeno obliko materiala, obenem pa lahko zagotovi tudi moč in trdoto pločevine.

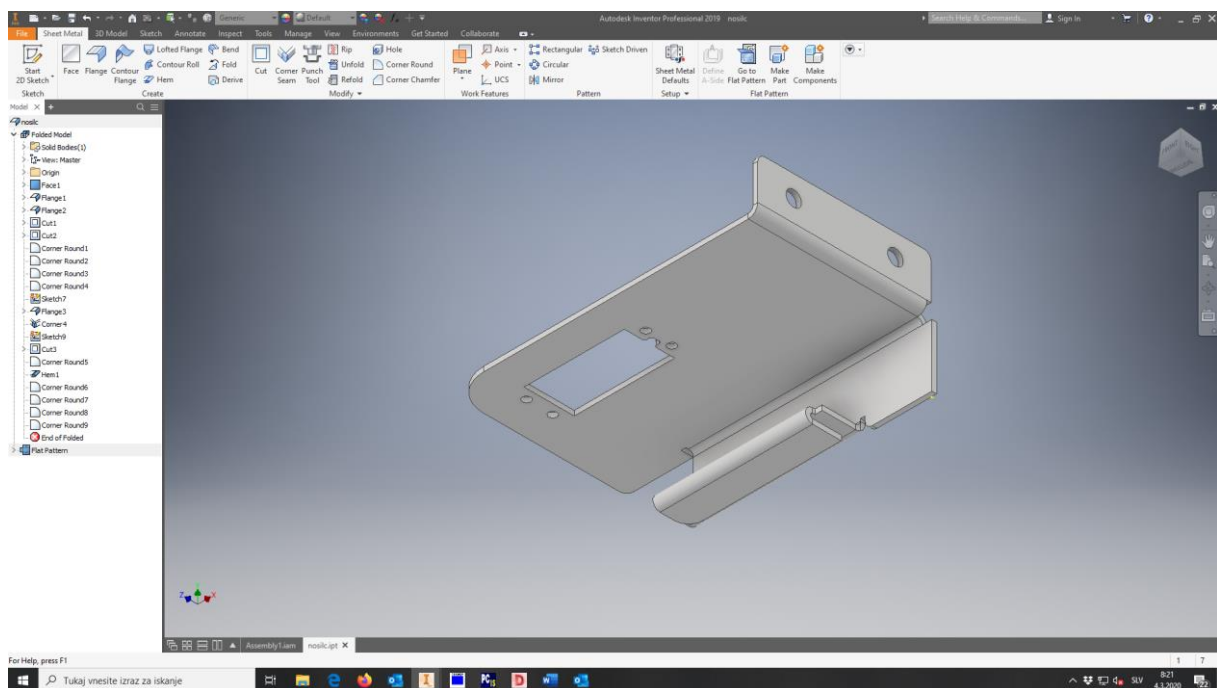
V nalogi sva ta postopek upogibanja kovine uporabila pri izdelavi ohišja ter nosilca klešč in servomotorja.



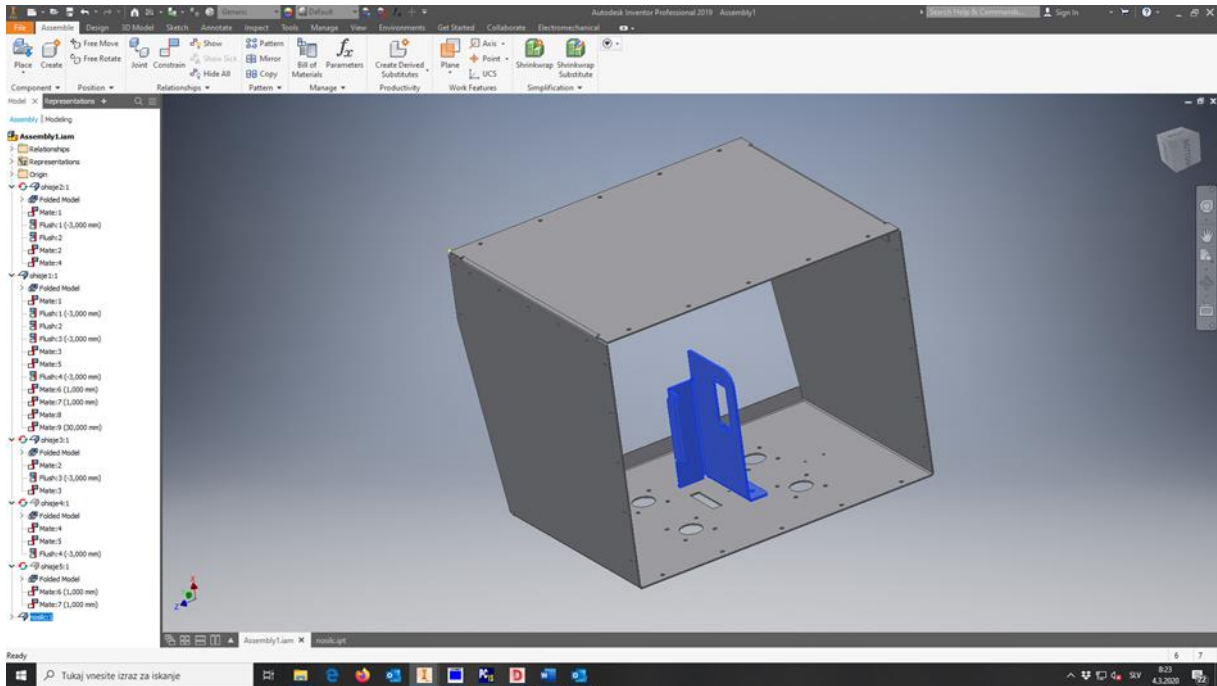
Slika 13: Stroj za upogibanje pločevine



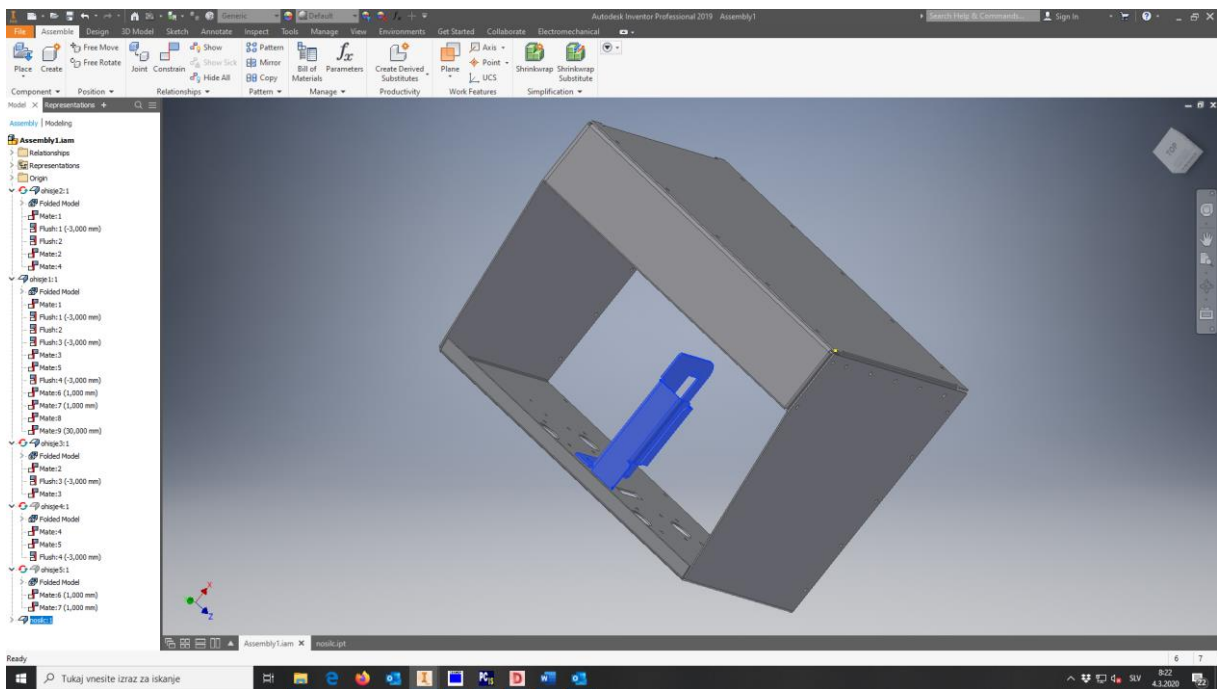
Slika 14: 3D prikaz ohišja



Slika 15: 3D prikaz nosilca



Slika 16: 3D prikaz pozicije nosilca



Slika 17: 3D prikaz pozicije nosilca

3.1.3.3 Struženje

Pri struženju rabimo stroj, ki se imenuje stružnica. V stružnico vpnemo obdelovanec, ki ga rezilo oblikuje z odstranjevanjem odvečnega materiala. Obdelovanec je vnaprej pripravljen kos, ki je pritrjen na stružnico in se z veliko hitrostjo lahko vrti okoli svoje osi. Nož za struženje, ki reže obdelovanec, je običajno točkoven (čeprav poznamo tudi večtočkovne, ki hkrati obdelujejo različne točke obdelovanca) in pritrjen na stružnico. Rezilo se dotakne in zažre v rotirajoč obdelovanec, odstrani odvečno plast materiala in postopoma določi končno želeno obliko izdelka. Po smeri podajanja orodja glede na os obdelovanca se struženje deli na vzdolžno in čelno, glede na mesto obdelave pa na notranje in zunanje struženje.

V podjetju KO-Lovec so nam na CNC-stružnici DOOSAN Lynx 220LY postružili puše, na katere sva namestila kolesa in jih nato pritrdila na osi koračnih motorjev.



Slika 18: Stružnica



Slika 19: Kolo s pušo

3.1.3.4 Vrtanje, grezenje in vrezovanje navojev

Vrtanje je postopek obdelave, kjer izdelujemo izvrtine v poln material ali pa povečujemo že obstoječe izvrtine.

Grezenje je širjenje že obstoječe izvrtine. Ločimo grobo grezenje, pri katerem je slabša kakovost izvrtine, fino grezenje, pri katerem je izvrtina bolj natančna, gladka in kakovostna, ter oblikovno grezenje, kjer valjasto izvrtino preoblikujemo v stopničasto, stožčasto izvrtino.

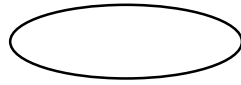
Za strojno vrezovanje navojev, predvsem v serijski proizvodnji, uporabljamo enote za vrezovanje navojev (svedri), ki imajo vgrajeno pinolo, ta pa določi korak navoja. Pomembno je, ali režemo navoj v prehodno ali neprehodno izvrtino.

Pri načrtovanju ohišja naprave smo pozicije lukenj že vnaprej določili z luknjami $\varnothing 2$, ki jih je izrezal laser. Nato sva luknje le povečala na $\varnothing 4$, pogreznila in v nekatere še vrezala navoj.

3.1.4 Vizualni prikaz programa

Preden sva začela s programiranjem, sva morala pripraviti načrt. S tem sva dobila nekakšen vpogled v strukturo programa in sva lažje opazila, kako narediti kakšno izboljšavo v programu ali kako odpraviti neko napako. To bi lahko storila tudi z Ishikawo ribjo kost, ki je boljša za analizo problematike, vendar sva se odločila uporabiti diagram poteka ali »flowchart«, ki smo se ga učili izdelovati pri pouku. S pomočjo diagrama poteka sva naredila grafični prikaz vseh korakov v programu. Diagram poteka predstavi program oz. proces s pomočjo grafičnih simbolov, ki jih je potrebno pri sestavi poznati. Najbolj osnovni in pogosti elementi so:

- Začetek in konec programa, t. i. terminatorji so predstavljeni z elipso.



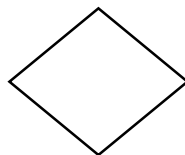
- Računske operacije ali podprogrami so predstavljeni s pravokotniki.



- Vhodne in izhodne operacije (branje ali pisanje) so predstavljene s paralelogrami.

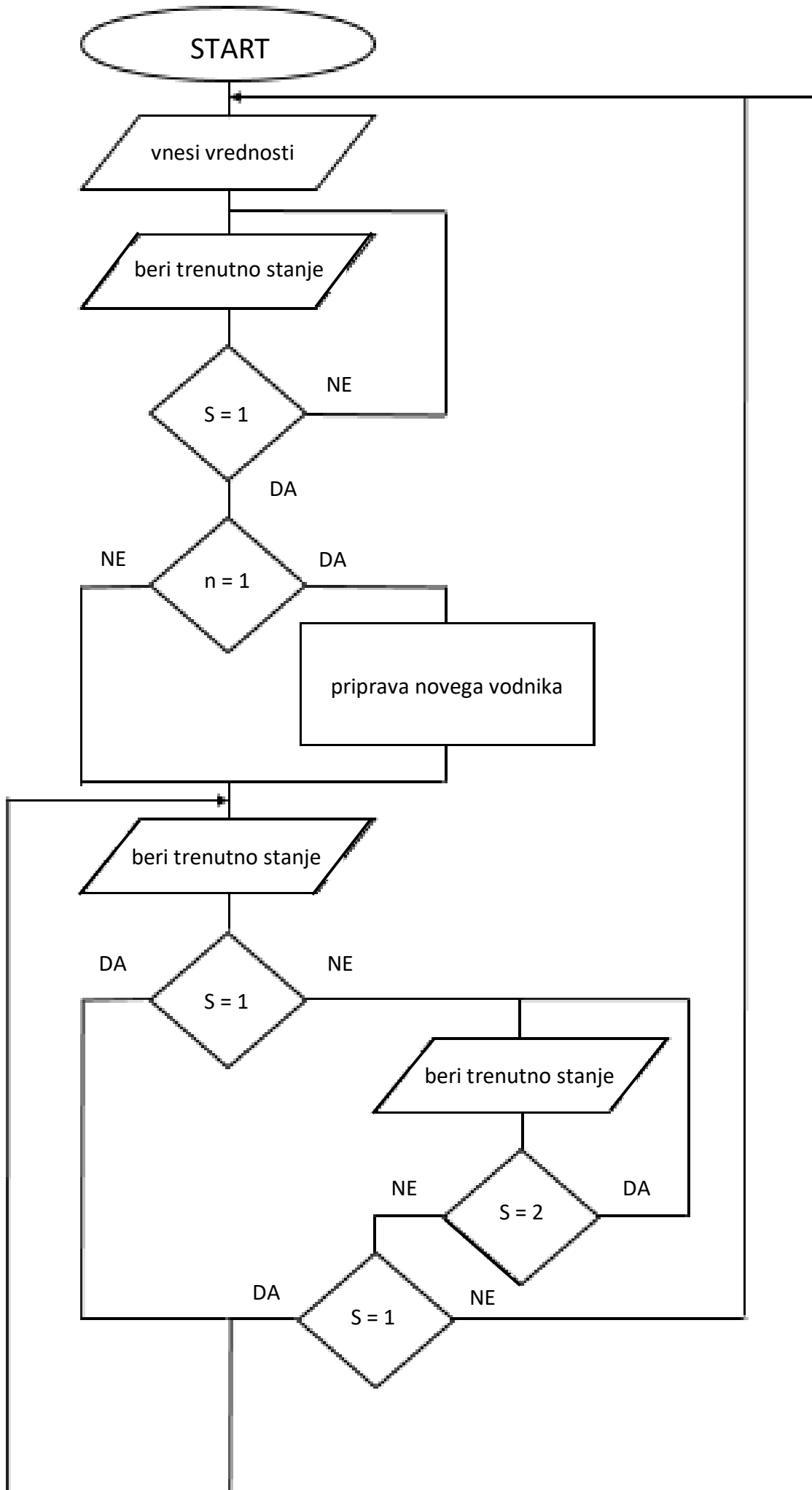


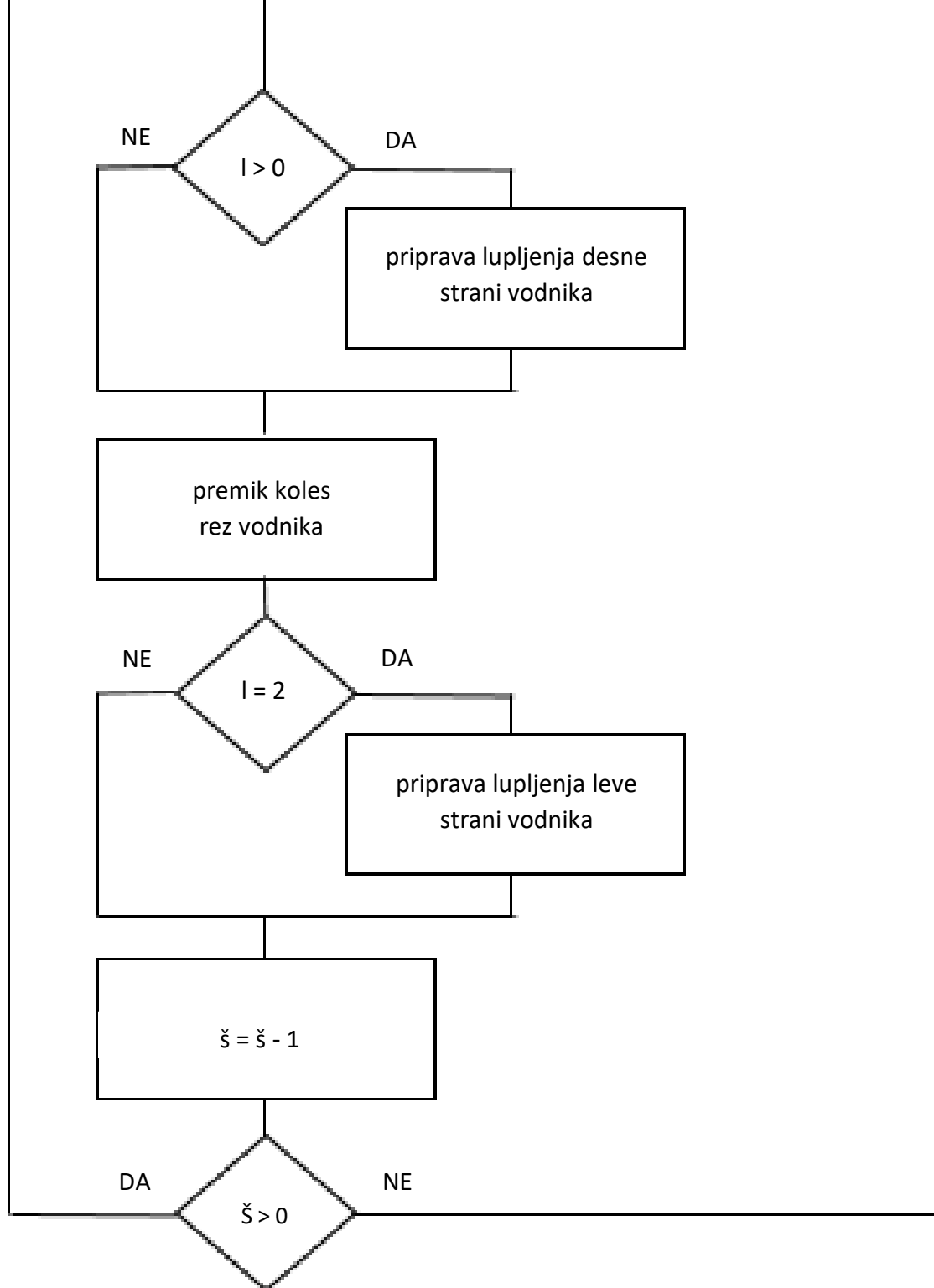
- Odločitve ali vejitve vsebujejo pogoj in dva izhoda ter predstavljata ali pogoj drži ali ne. Odločitve so predstavljene z rombi.



- Smer izvajanja je predstavljena s puščico, ki jo lahko pri risanju izpustimo, če je smer jasna (navzdol).







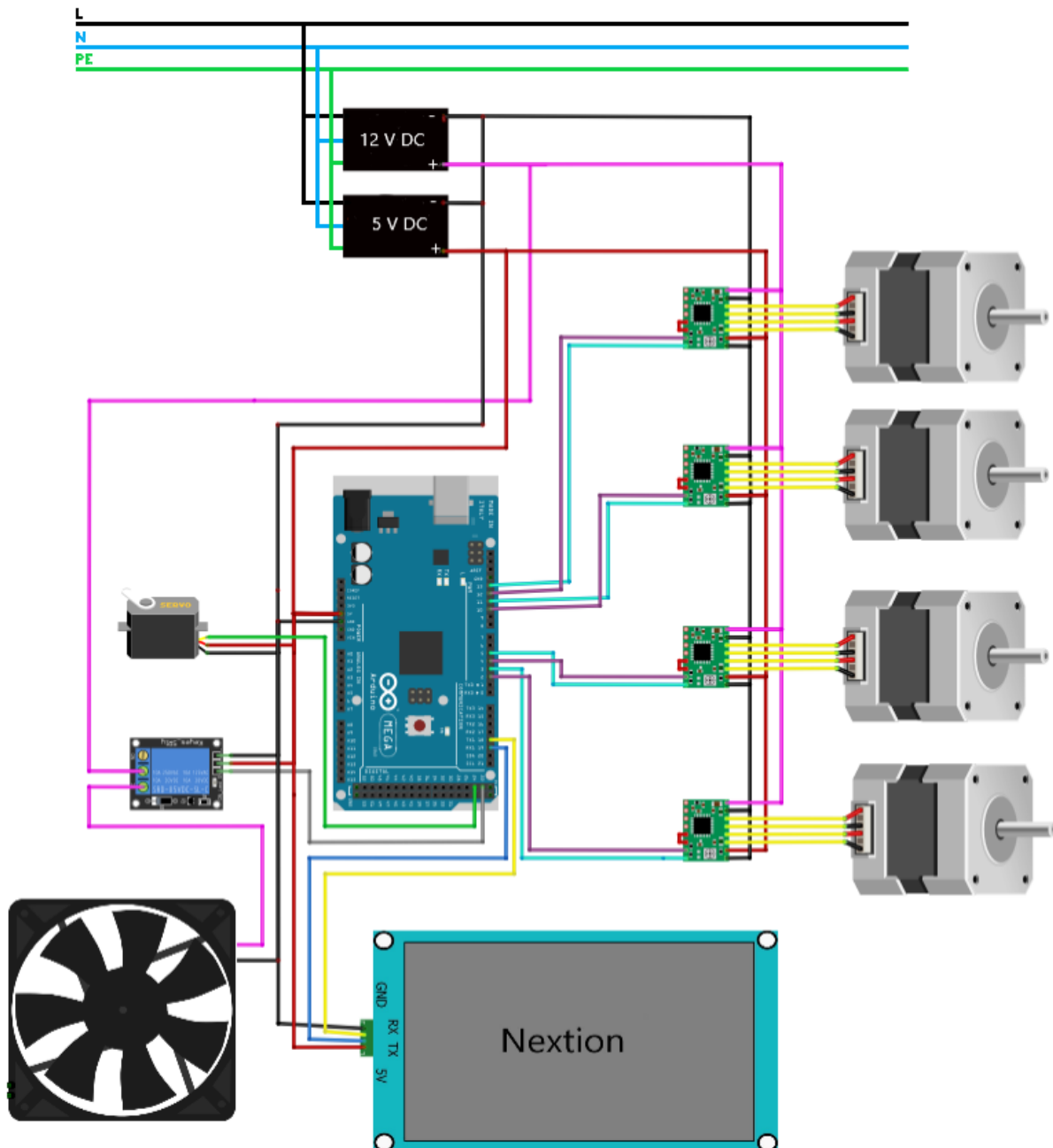
Slika 20: Diagram poteka

Na začetku s pomočjo zaslona na dotik vnesemo podatke, kot so:

- spremenljivka n , ki predstavlja, ali je vodnik nov ($n = 1$) ali je že bil v procesu in je konec vodnika že pri kleščah, kjer se začne meriti dolžina ($n = 0$),
- dolžina vodnika, ki jo program bere pod spremenljivko d ,
- število kosov, ki je predstavljeno s spremenljivko \check{s} ,
- stran lupljenja vodnika pod spremenljivko l , kjer $l = 0$ predstavlja, da se vodnik ne olup; $l = 1$, da se vodnik olup le na eni strani (desni) in $l = 2$, kjer se olupita obe strani (desna in leva).

Program se odziva odvisno od vnesenih spremenljivk, kakor je predstavljeno v diagramu poteka. V primeru, da nam kakšna spremenljivka ne ustreza, lahko program med delovanjem ustavimo in jo zapišemo znova ter s tem spremenimo le tisti del programa, ki je odvisen od spremenjene spremenljivke.

3.1.5 Vežalni načrt

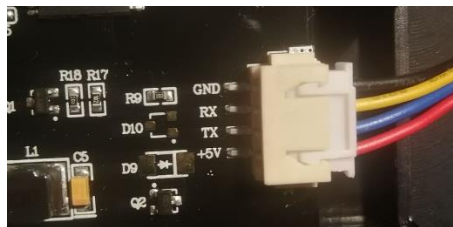


Slika 21: Vežalna shema

Ob prejetju vseh potrebnih komponent sva z nekaterimi dobila tudi vezalni načrt, ostale sva poiskala na spletu. Ko sva preučila posamezne vezalne načrte, sva izdelala celotnega, ki povezuje vse komponente med sabo (Slika 20).

Naprava bo priključena na omrežno napetost 230 V AC, ki jo bosta napajalnika pretvorila v svoji nazivni napetosti (5 V DC in 12 V DC). S 5 V napajalnikom se napajajo Arduino, Nextion, servomotor in gonilniki koračnih motorjev. Ostale komponente, kot so rele, ventilator ter gonilniki, napaja 12 V napajalnik.

Arduina napajamo preko 5 V nožice, pri čemer moramo paziti, da nanj ne pride večja napetost, saj lahko Arduino uniči, ker s takšno vezavo obidemo zaščitno diodo kot tudi varovalko, zato je posebej potrebna previdnost.



Slika 22: Nextion - konektor

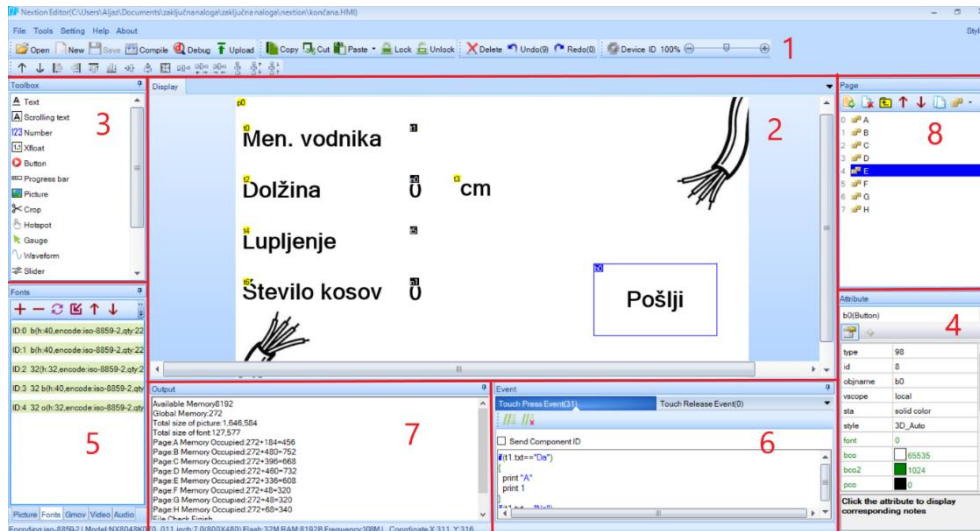
Priključitev Nextiona na Arduino je zelo enostavna. Narediti sva morala le štiri povezave: GND – minus, RX – sprejemni vodnik, TX – oddajni vodnik, in 5 V – potrebna napetost. Za pomoč pri pravilni vezavi so oznake napisane ob konektorju (Slika 21). Zaslona Nextion lahko povežemo tudi neposredno z nožico 5 V na Arduino, vendar ni priporočljivo. Delo z nezadostnim napajanjem lahko poškoduje zaslon, zato sva morala uporabiti zunanji vir napajanja.

Ostale komponente sva povezala, kot je razvidno na sliki. Pri ostali vezavi sva morala biti prav tako pozorna na pravilno priključitev.

3.1.6 Programiranje

3.1.6.1 Programiranje zaslona

Za programiranje Nextion zaslona sva uporabil program Nextion editor. Program vsebuje:



Slika 23: Nextion Editor

1 Meni in orodne vrstice – z njimi je možno shranjevanje programa, preverjanje delovanje programa, izvoz programa na krmilnik in uporaba določenih orodij.

2 Grafično oblikovanje programa – v tem delu se grafično oblikuje strani.

3 Podokno komponent – v njej izbiramo komponente, ki jih želimo grafično prikazati na ekranu.

4 Nastavitev komponent – v tem delu se vidijo podatki o komponenti, ki smo jo dodali, in spreminjamo grafične lastnosti komponente.

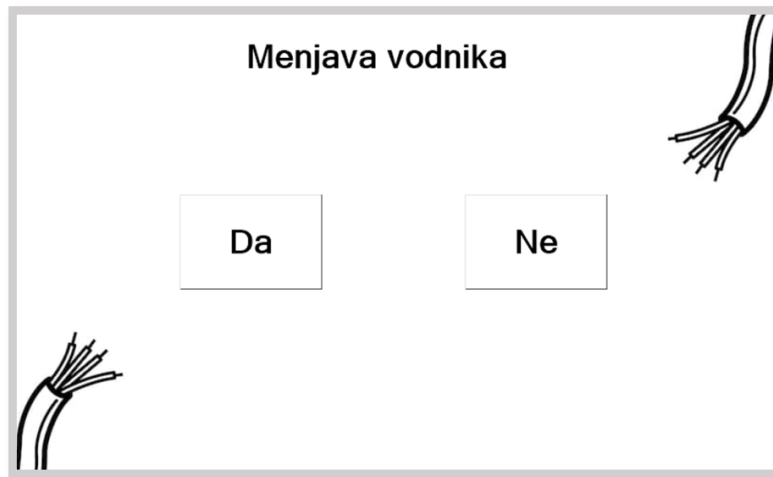
5 Slike in pisave – vsako pisavo in sliko, ki jo želimo uporabiti v programu, moramo posebej vnest v program, paziti moramo, da so slike prave velikosti.

6 Programsko okno – vanj zapišemo kodo, kaj želimo storiti, ko aktiviramo komponento.

7 Izhodno podokno – prikazani so podatki o programu in ob preverjanju program zazna, če so kje kakšne nepravilnosti.

8 Strani – v tem delu dodajamo nove strani in se premikamo med njimi.

Ko sva imela program zaključen, sva povezala krmilnik Arduino z računalnikom preko USB vhoda, nato pa še Arduina in Nextion preko vhodov Tx in Rx. Ko je bilo vse pravilno priključeno, sva s tipko »upload« na Nextion editerju prenesla program na zaslon, kjer se je po končanem nalaganju pokazala grafika, ki sva jo oblikovala.



Slika 24: Nextion - prva stran

Prvo stran sva sprogramirala tako, da s tipkama DA in NE izberemo, ali želimo v proces vstaviti nov vodnik ali nadaljujemo z uporabo starega. Ob pritisku na željeno tipko se pomaknemo na stran dve.



Slika 25: Nextion - druga stran

Na drugi strani sva morala narediti tipkovnico, v katero lahko vnašamo natančne vrednosti dolžin vodnika. Maksimalna vrednost, ki jo lahko vnesemo, je 999. S tipko DEL lahko vrednost izbrišemo in nastavimo novo. Ko smo z vrednostjo zadovoljni, pritisnemo tipko OK in se prestavimo na naslednjo stran. V primeru, da je vrednost manjša od 6, nas program ne spusti na naslednjo stran.



Slika 26: Nextion - tretja stran

Na tretji strani nastavljam, ali želimo vodnik olupiti na eni strani, na obeh ali pa ga sploh ne želimo lupiti. Ko pritisne željeno tipko, se prestavimo na stran, ki sledi.



Slika 27: Nextion - četrta stran

Na četrti strani je podobna tipkovnica kot na strani 2, le da tukaj vnašamo število kosov z maksimalno možno vrednostjo 99. Če se želimo premakniti na naslednjo stran, moramo pritisniti tipko OK, vnesena vrednost mora biti večja od 0.

Men. vodnika	Ne
Dolžina	33
Lupljenje	Brez
Število kosov	3

Pošlji





Slika 28: Nextion - peta stran

Na peti strani so prikazani vsi podatki, ki smo jih prej nastavili. Če s katerim podatkom nismo zadovoljni, kliknemo na vrednost, ki je napačna. S pritiskom na napačno vrednost nas pošlje na stran, kjer se ta podatek nastavlja in tam nastavimo novo vrednost. Ko smo zadovoljni z vsemi vrednostmi, pritisnemo tipko POŠLJI in pošlje vse podatke na Arduina.

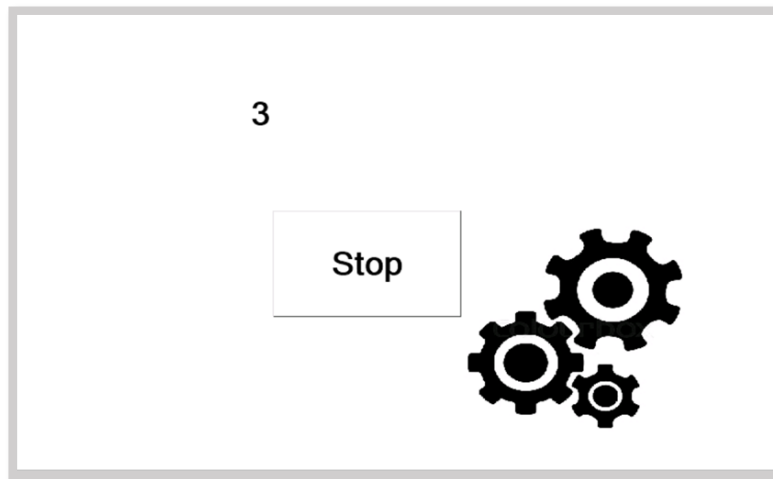
←

Start



Slika 29: Nextion - šesta stran

Na šesti strani je tipka START in tipka za vrnitev na prejšnjo stran. Ko smo prepričani, da smo vse podatke pravilno vnesli in je stroj primeren za uporabo, pritisnemo tipko START, ki požene stroj.



Slika 30: Nextion - sedma stran

Na sedmi strani lahko s tipko STOP zaustavimo program na Arduino, če ugotovimo, da kaj ne deluje v redu. Na strani je tudi polje, ki kaže, koliko kosov je še do konca programa. V primeru, da programa ne zaustavljamo, bo ob koncu delovanja Nextion avtomatsko preklopil na prvo stran, kjer mu ponovno nastavimo vrednosti.



Slika 31: Nextion - osma stran

Na osmo stran pridemo samo v primeru, da smo na prejšnji strani pritisnili tipko STOP. Na tej strani sta tipka NADALJUJ in tipka POENOSTAVI. V primeru, da pritisnemo tipko NADALJUJ, nas Nextion pošlje na prejšnjo stran in nadaljuje program na Arduino. Če pritisnemo tipko POENOSTAVI, nas Nextion prestavi na prvo stran in zaključi s programom.

3.1.6.2 Programiranje mikrokrmilnika

Za programiranje mikrokrmilnika Arduino sva uporabila program Arduino IDE. Pri pisanju programa sva uporabljala programski jezik C++. V pomoč pri pisanju nama je bil diagram poteka, saj sva okvirno vedela, kako naj bi program izgledal. Za vsako komponento posebej sva morala na spletu poiskati, katere ukaze morava uporabiti za kontroliranje komponente. Ko sva imela program za posamezno komponento narejen, sva ga testirala in vnesla vrednosti.

3.1.6.2.1 Programiranje koračnih motorjev

```
const int stepPin = 12;
const int dirPin = 13;

void setup() {
  pinMode(stepPin, OUTPUT); //
  pinMode(dirPin, OUTPUT);
}

void loop() {
  digitalWrite(dirPin, LOW);
  for(int x = 0; x < 200; x++) {
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(1000);
  }
  delay(2000);
}
```

Slika 32: Program - koračni motor

Za koračne motorje najprej nastavimo vrednosti spremenljivkama stepPin in dirPin. Ti vrednosti nam povesta, na katere nožice krmilnika moramo priključiti gonilnike motorja. V void setupu definiramo, da so nožice izhodi in na katerem mestu so. V void loopu, z dirPinom nastavljamo smer vrtenja motorja (HIGH,LOW), s for zanko pa povemo, za koliko se bo motor zavrte (200 enot je en obrat). Obrati motorja se izvajajo s stepPinom.

3.1.6.2.2 Programiranje Nextiona

```
int stevec = -1;
static int A = 0, B=0, C = 0, D = 0;
char buffer_izpis[50];
static byte msg[30], status_=0;
static byte i=0,k=0,msg_call=0,status_change=0;

void serialEvent1(){

  msg[i]=Serial1.read();
  Serial.println(msg[i]);
  if(msg[0]==1){
    status_=1;
    i=0;
    status_change=1;
    return;
  }else if(msg[0]==2){
    status_=2;
    i=0;
    status_change=1;
    return;
  }else if(msg[0]==3){
    status_=3;
    status_change=1;
    i=0;
    return;
  }

  if(msg[0]=='A'){
    if(msg[i]!=';'){
      i++;
    }
    else{
      i=0;
      msg_call=1;
    }
  }
}

void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
  Serial.println("ok");
}

void loop() {
  if(msg_call){
    A=msg[1];
    B=msg[6];
    B+=msg[7]*256;
    C=msg[11];
    D=msg[16];
    msg_call=0;

  }

  if(status_change){
    status_change=0;
  }
}
```

Slika 33: Program - Nextion 1

V tem delu programa je urejena komunikacija med Nextionom in Arduino. Ko Nextion pošlje podatke Arduino, ta ugotovi, za katero vrednost gre in to vrednost nato shrani kot spremenljivke.

```
void loop() {
goto_page(0);
printNxt();
}

void printNxt() {
Serial1.print(F("t11.txt=\"));
Serial1.print(stevilo);
Serial1.print(F("\"));
Serial1.write(0xff);
Serial1.write(0xff);
Serial1.write(0xff);
}

void goto_page(int gotopage) {
Serial1.print("page ");
Serial1.print(gotopage);
//Serial1.print("\xFF\xFF\xFF"); // 1
Serial1.write(0xff);
Serial1.write(0xff);
Serial1.write(0xff);
}
```

Slika 34: Program - Nextion 2

V program sva vključila funkcijo printNxt (), s katero pošiljamo število kosov, ki jih je stroj že napravil. S funkcijo goto_page (0) nas po končanem procesu vrne na prvo stran.

3.1.6.2.3 Programiranje servomotorja

```
#include <Servo.h>
Servo Servo1;
int koncnipolozajsrvota = 120;
int servorezanje = 15;

void setup() {
  Servo1.attach(24);
}

void loop() {
  Servo1.write(koncnipolozajsrvota);
  delay(1000);
  Servo1.write(servorezanje);
  delay(1000);
  Servo1.write(koncnipolozajsrvota);
  delay(1000);
}
```

Slika 35: Program - Servomotor

Izhod za servomotor sva določila s pin 24. Program deluje tako, da so vrednosti, ki so podane v spremenljivkah, stopinje, do katerih se bo motor odklonil.

3.1.6.2.4 Program za ventilator

```
void setup() {
  pinMode(22, OUTPUT);
  digitalWrite(22, HIGH);
}

void loop() {
  digitalWrite(22, LOW);
  delay(4000);
  digitalWrite(22, HIGH);
  delay(4000);
}
```

Slika 36: Program - ventilator

Za krmiljenje ventilatorja uporabimo rele. Z Arduinom nato krmilimo rele preko 22 izhoda, ki ga lahko poljubno vklopimo (`digitalWrite(22,HIGH)`) in izklopimo (`digitalWrite(22,LOW)`).

3.1.6.2.5 Združen program

```
#include <Servo.h>
Servo Servo1;
long izracun = 0;
int koncnipolozajsvota = 120;
int servorezanje = 15;
int servoluoljenje = 40;
int stevec = -1;
static int A = 0, B=0, C = 0, D = 0;
char buffer_izpis[50];
static byte msg[30], status_=0;
static byte i=0,k=0,msg_call=0,status_change=0;
int stevilo = 0;
const int stepPin = 12;
const int dirPin = 13;
const int stepPin1 = 10;
const int dirPin1 = 11;
const int stepPin2 = 4;
const int dirPin2 = 5;
const int stepPin3 = 2;
const int dirPin3 = 3;

void serialEvent1(){

  msg[i]=Serial1.read();
  //Serial.println(i);
  Serial.println(msg[i]);
  if(msg[0]==1){
    status_=1;
    i=0;
    status_change=1;
    return;

  }else if(msg[0]==2){
    status_=2;
    i=0;
    status_change=1;
    return;

  }else if(msg[0]==3){
    status_=3;
    status_change=1;
    i=0;
    return;

  }
  if(msg[0]=='A'){
    if(msg[i]!=';'){
      i++;
    }
    else{
      i=0;
      msg_call=1;
      //Serial.println("fend");
    }
  }
}
```

```
void setup() {

  Serial.begin(9600);
  Serial1.begin(9600);
  Serial.println("ok");

  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
  pinMode(stepPin1, OUTPUT);
  pinMode(dirPin1, OUTPUT);
  pinMode(stepPin2, OUTPUT);
  pinMode(dirPin2, OUTPUT);
  pinMode(stepPin3, OUTPUT);
  pinMode(dirPin3, OUTPUT);
  Servo1.attach(24);
  pinMode(22, OUTPUT);

}

void loop() {

  if(msg_call){
    A=msg[1];
    B=msg[6];
    B+=msg[7]*256;
    C=msg[11];
    D=msg[16];
    msg_call=0;

    Serial.print("A:");Serial.print(A);Serial.print(" ");//Serial.println();
    Serial.print("B:");Serial.print(B);Serial.print(" ");//Serial.println();
    Serial.print("C:");Serial.print(C);Serial.print(" ");//Serial.println();
    Serial.print("D:");Serial.print(D);Serial.print(" ");Serial.println();
  }

  if(status_change){

    Serial.print("status change:");Serial.println(status_);
    status_change=0;

  }

  if(status_==1){
    printNxt();

    digitalWrite(22, HIGH);

    if(A==1){
    digitalWrite(dirPin, HIGH);
    digitalWrite(dirPin1, LOW);
    for(int x = 0; x < 100; x++) {
      digitalWrite(stepPin, HIGH);
      digitalWrite(stepPin1, HIGH);
      delayMicroseconds(1000);
      digitalWrite(stepPin, LOW);
      digitalWrite(stepPin1, LOW);
      delayMicroseconds(1000);
    }
  }

  delay(2000);
```

```
digitalWrite(dirPin,LOW);
digitalWrite(dirPin1,HIGH);
for(int x = 0; x < 100; x++) {
    digitalWrite(stepPin,HIGH);
    digitalWrite(stepPin1,HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin,LOW);
    digitalWrite(stepPin1,LOW);
    delayMicroseconds(1000);
}

Servo1.write(koncnipolozajsvota);
delay(1000);
Servo1.write(servorezanje);
delay(1000);
Servo1.write(koncnipolozajsvota);

for(int i=0 ; i<D; i++){

    if(C>0) {

        digitalWrite(dirPin,LOW);
        digitalWrite(dirPin1,HIGH);
        for(int x = 0; x < 200; x++) {
            digitalWrite(stepPin,HIGH);
            digitalWrite(stepPin1,HIGH);
            delayMicroseconds(1000);
            digitalWrite(stepPin,LOW);
            digitalWrite(stepPin1,LOW);
            delayMicroseconds(1000);
        }

        Servo1.write(koncnipolozajsvota);
        delay(1000);
        Servo1.write(servoluoljenje);
        delay(1000);

        digitalWrite(dirPin,HIGH);
        digitalWrite(dirPin1,LOW);
        for(int x = 0; x < 200; x++) {
            digitalWrite(stepPin,HIGH);
            digitalWrite(stepPin1,HIGH);
            delayMicroseconds(1000);
            digitalWrite(stepPin,LOW);
            digitalWrite(stepPin1,LOW);
            delayMicroseconds(1000);
        }
        Servo1.write(koncnipolozajsvota);
        delay(1000);

    }

    izracun = (B/25.76)*200;
    Serial.print("izracun");
    Serial.print(izracun);
```

```
digitalWrite(dirPin,HIGH);
digitalWrite(dirPin1,LOW);
digitalWrite(dirPin2,HIGH);
digitalWrite(dirPin3,LOW);
for(int x = 0; x < izracun; x++) {
digitalWrite(stepPin,HIGH);
digitalWrite(stepPin1,HIGH);
digitalWrite(stepPin2,HIGH);
digitalWrite(stepPin3,HIGH);
delayMicroseconds(1000);
digitalWrite(stepPin,LOW);
digitalWrite(stepPin1,LOW);
digitalWrite(stepPin2,LOW);
digitalWrite(stepPin3,LOW);
delayMicroseconds(1000);
}

Servo1.write(koncnipolozaajsvota);
delay(1000);
Servo1.write(servorezanje);
delay(1000);
Servo1.write(koncnipolozaajsvota);
delay(1000);

if(C==2){

digitalWrite(dirPin,LOW); // low high
digitalWrite(dirPin1,HIGH);
digitalWrite(dirPin2,LOW); // low high
digitalWrite(dirPin3,HIGH);
for(int x = 0; x < 200; x++) { //nast
digitalWrite(stepPin,HIGH);
digitalWrite(stepPin1,HIGH);
digitalWrite(stepPin2,HIGH);
digitalWrite(stepPin3,HIGH);
delayMicroseconds(1000);
digitalWrite(stepPin,LOW);
digitalWrite(stepPin1,LOW);
digitalWrite(stepPin2,LOW);
digitalWrite(stepPin3,LOW);
delayMicroseconds(1000);
}

Servo1.write(koncnipolozaajsvota);
delay(1000);
Servo1.write(servoluoljenje);
delay(2000);

digitalWrite(dirPin,LOW); // low high
digitalWrite(dirPin1,HIGH);
digitalWrite(dirPin2,LOW); // low high
digitalWrite(dirPin3,HIGH);
for(int x = 0; x < 200; x++) { //nasta
digitalWrite(stepPin,HIGH);
digitalWrite(stepPin1,HIGH);
digitalWrite(stepPin2,HIGH);
digitalWrite(stepPin3,HIGH);
delayMicroseconds(1000);
digitalWrite(stepPin,LOW);
digitalWrite(stepPin1,LOW);
digitalWrite(stepPin2,LOW);
digitalWrite(stepPin3,LOW);
delayMicroseconds(1000);
}
```

```
Servo1.write(koncnipolozajsrvota);
delay(1000);

}

stevilo = stevilo+1;
printNxt();

}
goto_page(0);
status_=0;
stevilo=0;
digitalWrite(22,LOW);
}
}

void printNxt(){
  Serial1.print(F("t11.txt="));
  Serial1.print(stevilo);
  Serial1.print(F("\n"));
  Serial1.write(0xff);
  Serial1.write(0xff);
  Serial1.write(0xff);
}

void goto_page(int gotopage){
  Serial1.print("page ");
  Serial1.print(gotopage);
  //Serial1.print("\xFF\xFF\xFF");
  Serial1.write(0xff);
  Serial1.write(0xff);
  Serial1.write(0xff);
}
```

Slika 37: Program - združen

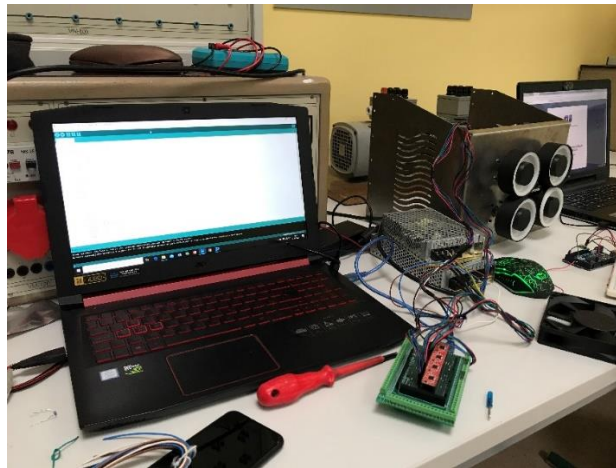
3.1.7 Testiranje

Na začetku sva vsako posamezno komponento pravilno povezala, zanjo napisala program in testirala, tako da sva dobila željeni rezultat.

Pri programiranju Nextiona sva prišla do težav, saj nisva našla veliko uporabnih informacij glede programiranja. Progame sva nato združila, tako da je deloval, kot je prikazano v diagramu poteka (Slika 19). Nato sva vse komponente zvezala, preverila, če je vezava pravilna, in priključila na napajalnik. Pojavile so se težave z gonilniki, saj je prišlo do pregrevanja in so prenehali delovati. Veliko jih ni delovalo že ob prejetju, kar sva opazila že pri testiranju komponent, zato se je čas testiranja in izdelave drastično podaljšal.

Na začetku sva testirala napravo brez vodnika, da sva preverila, če naprava deluje tako kot mora.

Ko sva testirala z vodnikom, je prišlo do težav, saj so bili koračni motorji preblizu. To težavo bova v nadaljevanju skušala odpraviti. Predvideva, da bo potrebno popraviti ohišje naprave, tako da bodo koračni motorji bolj oddaljeni.



Slika 38: Testiranje

Po testiranju smo začeli z iskanjem novih možnosti za nadgradnjo stroja, ki smo jih predlagali podjetju KO-Lovec. Napravo bi lahko nadgradili še z/s:

- nastavitvijo koračnih motorjev in škarij za več prerezov vodnikov (z vzmetjo),
- samodejnim izbiranjem in vstavljanjem vodnikov,
- samodejnim čiščenjem odrezkov (pnevmatsko) v koš,
- možnostjo namestitve zaključkov (tulci, fastoni, spojke itd.),

3.2 RAZPRAVA

Določene teze nisva mogla preveriti, saj naprava še ne deluje tako kot mora, zato bova te teze lahko preverila šele, ko odpraviva vse napake. Testiranje se je prestavilo zaradi težav z gonilniki, ki so se preobremenili, in zaradi aktualnih težav s pošiljanjem s Kitajske niso prišli pravočasno.

Program sva zastavila tako, da deluje hitreje, kot bi potekalo ročno delo delavca. Te teze še nisva mogla preveriti v praksi.

Napravi vnašamo podatke preko zaslona na dotik, ki jih mikrokrmilnik bere kot spremenljivke, kar delujejo tako, kot sva si želela.

Naprava ne zavzema veliko prostora, kar pomeni, da je lahko nameščena na delovni mizi in bo delavcu ostalo dovolj prostora za nemoteno delovanje, po potrebi pa jo lahko tudi prestavimo.

Skupna vrednost nabavljenih sestavnih delov naprave je 400 €, medtem ko najinega vloženega dela in delov, ki zaradi tehničnih napak niso uporabni, nisva posebej ovrednotila. To ceno bova lahko v praksi preverila šele, ko bo naprava delovala. Vendar predvideva, da se bo cena povrnila v treh mesecih uporabe, saj ni potrebe po plačevanju delavca, ki bi to delo opravljal ročno.

4 ZAKLJUČEK

Med izdelavo raziskovalne naloge sva se veliko naučila, predvsem na področju strojništva (konstruiranje naprave) in programiranja nadzornega panela ter Arduina, saj imava na tem področju manj izkušenj. Zadala sva si kar zahtevno nalogo, imela sva številne težave s konstruiranjem in programiranjem, ki sva jih rešila tako, da sva se posvetovala z mentorji, zaposlenimi v podjetju KO-Lovec, s starši in s sošolci.

Večino podatkov sva našla na spletu.







1. Naprava izvaja avtomatsko rezanje in lupljenje vodnikov prereza 0,5 mm ²	
2. Proces poteka dvakrat hitreje kot ročno delo.	
3. Posluževanje naprave se izvaja preko zaslona na dotik.	
4. Naprava je v celoti mobilna.	
5. Stroški naprave se povrnejo v enem letu.	
6. Naprava je varna pri delu.	

Tabela 1: Ovrednotenje hipotez

5 VIRI IN LITERATURA

- 1) Mikrokrmilnik Arduino. Pridobljeno 27. 2. 2020 s strani:

<https://www.arduino.cc/>.

- 2) Nextion. Pridobljeno 28. 2. 2020 s strani:

<https://nextion.tech/> .

- 3) Vezava Nextiona. Pridobljeno 28. 2. 2020 s strani:

<https://randomnerdtutorials.com/nextion-display-with-arduino.>

<https://www.youtube.com/watch?v=mdkUBB60Hol>.

- 4) Koračni motor. Pridobljeno 28. 2. 2020 s strani:

https://en.wikipedia.org/wiki/Stepper_motor.

- 5) Gonilniki. Pridobljeno 29. 2. 2020 s strani:

<https://www.pololu.com/>.

- 6) Servomotorji. Pridobljeno 1. 3. 2020 s strani:

www.ps-log.si/produkti/servo-motorji.

- 7) Razrez in upogibanje pločevine. Pridobljeno 1. 3. 2020 s strani:

<https://www.ko-lovec.si/>.

7 ZAHVALA

Zahvaljujeva se vsem, ki so kakorkoli pomagali pri izdelavi raziskovalne naloge. Še posebej bi se rada zahvalila najinemu mentorju Gregorju Kramerju, univ. dipl. inž. el. in Davorju Zupanu, inž. el., za ves trud, čas, podporo, vztrajnost in nasvete za izdelovanje raziskovalne naloge.

Prav tako bi se rada zahvalila kolektivu KO-Lovec za ideje, nasvete in tudi kritike pri izdelovanju izdelka, staršem in sošolcem, ter lektorici za pregled in popravo raziskovalne naloge.

Še enkrat vsa zahvala vsem, ki so pripomogli k nastanku te raziskovalne naloge, saj brez njihove pomoči naloga ne bi bila uspešna.

IZJAVA*

Mentor **Gregor Kramer** v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom **Naprava za avtomatsko rezanje in lupljenje vodnikov**, katere avtorji so **Leon Kovše, Aljaž Perc**:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 5. 6. 2020

žig šole

Podpis mentorja



Podpis odgovorne osebe



*

POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.