

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jani Černe

**Vodni slap kot tiskalnik**

DIPLOMSKO DELO  
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Dušan Kodek

Ljubljana, 2011

# IZJAVA O AVTORSTVU

## diplomskega dela

Spodaj podpisani      Jani Černe,  
z vpisno številko      63040020,

sem avtor diplomskega dela z naslovom:

Vodni slap kot tiskalnik

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Dušana Kodeka
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 13.9.2011

Podpis avtorja:



Št. naloge: 01753/2011

Datum: 01.04.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **JANI ČERNE**

Naslov: **VODNI SLAP KOT TISKALNIK**  
**WATERFALL AS A PRINTER**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Vodni slap v obliki elektronsko krmiljene padajoče vodne zavese omogoča nenavaden in zanimiv način za prikazovanje slik, besedil in poljubnih vzorcev. V svetu je v parkih in veleblagovnicah znanih več realizacij te vrste, ki so za obiskovalce zelo privlačne. Na osnovi znanih rešitev zasnujte vodni tiskalnik te vrste z 80 vodnimi šobami, ki naj bo sposoben prikazovati slike in besedila. V ta namen najprej definirajte format za shranjevanje bitnih slik. Nato na osnovi razvojnega orodja Arduino izdelajte vezje, ki v skladu s shranjenim bitnim zapisom krmili vodne šobe. Izberite primerne ventile za odpiranje-zapiranje šob ter poskrbite za električno varnost v vodnem okolju. Izdelajte ustrezno programsko opremo in na zgledih preverite pravilnost delovanja vodnega tiskalnika.

Mentor:

  
prof. dr. Dušan Kodek



Dekan:

  
prof. dr. Nikolaj Zimic

# Zahvala

Zahvaljujem se prof. dr. Dušanu Kodeku za mentorstvo, brez katerega bi projekt ostal le ideja. Posebna zahvala gre doc. dr. Mateju Možku s fakultete za elektrotehniko za strokovno pomoč pri izdelavi posameznih komponent in dragocene nasvete. Hvala prijateljem, ki ste mi v ključnih trenutkih priskočili na pomoč.

Hvala očetu za podporo.

*Diplomsko delo posvečam vsem z idejami v  
garaži.*

# Kazalo

<b>Povzetek</b> .....	<b>1</b>
<b>Abstract</b> .....	<b>2</b>
<b>1 Uvod</b> .....	<b>3</b>
<b>2 Program za risanje in Delphi</b> .....	<b>4</b>
2.1 Splošno .....	4
2.1.1 Razvojno okolje Delphi .....	4
2.1.2 Portable pixmap format (PPM) .....	5
2.2 Program za risanje .....	6
<b>3 Vodni tiskalnik</b> .....	<b>9</b>
3.1 Vezje .....	9
3.1.1 Arduino Mega2560.....	9
3.1.2 Pomikalni register .....	10
3.1.3 Optospojnik .....	14
3.1.4 Program za Arduino .....	15
3.2 Vodni segment.....	16
3.2.1 Rezervoar s šobami.....	17
3.2.2 Zapiralo šobe .....	18
3.2.3 Zbiralni rezervoar s potopno črpalko .....	19
3.3 Delovanje vodnega tiskalnika.....	21
<b>4 Sklepne ugotovitve</b> .....	<b>22</b>
<b>Dodatek A</b> .....	<b>24</b>
A.1 Slike vodnega tiskalnika .....	24
A.2 Seznam slik.....	28
<b>Dodatek B</b> .....	<b>29</b>
B.1 Shema elektronike.....	29
<b>Dodatek C</b> .....	<b>30</b>
C.1 Priložen CD.....	30
<b>Viri in literatura</b> .....	<b>31</b>

# Seznam uporabljenih kratic

RC snubber	(angl.) Resistor Capacitor, blažilnik z uporom in kondenzatorjem
PPM	(angl.) Portable PixMap, format slike
TXT	(angl.) Text File, datoteka z besedilom
ASCII	(angl.) American Standard Code for Information Interchange, ameriški standardni nabor za izmenjavo informacij
RGB	(angl.) »Red, Green, Blue«, osnovni barvni model
USB	(angl.) Universal Serial Bus, univerzalno serijsko vodilo
Piksel	(angl.) Pixel, točka
SIPO	(angl.) »Serial-in, Paralel-out«, zaporedni vhod, vzporedni izhod

# Povzetek

Z diplomskim delom sem želel realizirati računalniško krmiljen vodni tiskalnik, ki bi bil zamenjava običajnim slapovom v vodnih parkih, za popestritev pred hoteli ali kot zanimivost na sejmiščih. Podobni izdelki sicer že obstajajo, vendar jih pri nas še nisem zasledil, poleg tega pa zajema več področij, zaradi katerih se mi je ideja o izdelavi zdela zanimiva.

V prvi vrsti se posvečam izdelavi računalniškega programa, ki predstavlja uporabniški vmesnik in je namenjen upravljanju vodnega tiskalnika. Na kratko je predstavljeno razvojno okolje Delphi v katerem je bil izdelan program za risanje ter format slik PPM. Opisane so tudi tehnike pretvorbe slik in prikaz izhoda, ki je prvi rezultat treh segmentov. V nadaljevanju sledi opis elektronike, kot vmesnega člena med programom in vodnim segmentom. V tem segmentu se osredotočam na predstavitev uporabljenih elektronskih elementov ter njihovemu delovanju. Zadnji del je namenjen opisu delovanja vodnega segmenta ter načinu izdelovanja posameznih delov, ki jih vključuje.

## **Ključne besede:**

Vodni tiskalnik, Delphi, format slik PPM, elektronsko krmiljenje

# Abstract

Waterfall printers already exist but are not widely used because, beside the planing, the cost of the components is very high. With this thesis I wanted to make a home version of the waterfall printer with easily accessible components and then compare the results with a professional one.

This thesis describes what the waterfall printer is and how it works. The project consists of three main parts. The first part is a computer program developed in Delphi environment which represents the user interface to control the printer. Beside the functions of the program, it is explained how the PPM image format is made and how it is used. Second part is electronics that is needed to trigger the electromagnets. Electromagnets are embedded in the water reservoir which is the last part of the waterfall printer. Describing the mechanical parts of the water reservoir I show the methods used for manufacturing individual parts with possible ways of further development.

## **Key words:**

Waterfall printer, Delphi, PPM image format, electronic control

# Poglavje 1

## Uvod

V vsakodnevem življenju se srečujemo z oglaševalskimi panoji in svetlečimi tablami, ki pa znajo biti nerodno postavljene in včasih celo moteče. Zaradi lastnosti okolja pa nekaterih sploh ne moremo postaviti. Tako je v vodnih parkih mogoče opaziti običajne vodne slapove oz. vodne zaves, ki pa same po sebi nimajo nobenega posebnega učinka. V ta namen je bil izdelan vodni tiskalnik.

Vodni tiskalnik je elektronsko krmiljen vodni slap, ki je sposoben prikazovati binarne slike, vzorce in napise. Tako kot pri običajnem vodnem slapu, tudi pri vodnem tiskalniku voda pada v liniji vendar se namesto z enega izvora voda zliva s šob. Vsaka šoba je elektronsko krmiljena.

Vodni tiskalniki so v svetu znani. Eden prvih je nastal že v poznih sedemdesetih, katerega je izdelal prof. Stephen Pevnick z univerze v Wisconsinu, ZDA. Prvotni tiskalnik je bil oblike horizontalnega kvadrata s stranicami 1,2m in skupno 64 šobami. Na svetovnem spletu je mogoče zaslediti vrsto vodnih tiskalnikov postavljenih v nakupovalnih središčih [1] in avtomobilskih sejmih [2]. Pri v diplomu uporabljeni rešitvi sem se zgledoval po slikovnem materialu Pevnick Designa [3].

V tem delu predstavljen tiskalnik sestoji iz programskega, elektronskega ter vodnega segmenta. Programski segment skrbi za urejanje slike za izris in je izdelan v programskem jeziku Object Pascal s pomočjo razvojnega okolja Delphi. Program je samostojna aplikacija, katerega izhod je razumljiv elektroni, ki predstavlja naslednji segment. Elektronika sliko prebere in jo pretvori v električne impulze, ki krmilijo vodne ventile oz. zapirala šob. Voda se ob odprtju ventilov izlije z rezervoarja in ustvari sliko, nato pa se preko potopne črpalke vrne v rezervoar.

## **Poglavje 2**

# **Program za risanje in Delphi**

### **2.1 Splošno**

Končni namen vodnega tiskalnika je prikazovanje slik, napisov in raznih vzorcev. Na zgornjem robu okvirja tiskalnika so postavljene elektronsko krmiljene šobe, ki jih vklaplja razvojno orodje Arduino. Želena slika je potrebno nekako predstaviti, da jo elektronika razume. Ker je preklopna elektronika izdelana z 80 izhodi, je smiselna predstavitev slike ravno v tej resoluciji. Potrebno je bilo izdelati računalniški program s katerim lahko poljubno sliko določene resolucije spremenimo v zaporedje enic in ničel. Izhodna datoteka programa je torej slika enic in ničel, ki jih elektronika razume kot kapljice. Program je izdelan v programskem jeziku Object Pascal in je poleg pretvorbe namenjen tudi popravljanju slike ter risanju samem.

#### **2.1.1 Razvojno okolje Delphi**

Delphi [4] je razvojno okolje, ki temelji na objektivnem Pascalu. Zasnovali so ga pri podjetju Borland leta 1995 in je bil sprva namenjen razvijanju aplikacij v okolju Microsoft Windows za vodenje baz podatkov, evidenc in podobnih opravil. Baza podatkov v tem projektu ni uporabljena, je pa Delphi izbran zaradi enostavne uporabe in avtorjevih osebnih preferenc.

## 2.1.2 Portable pixmap format (PPM)

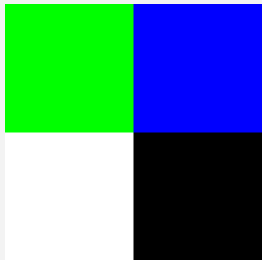
Format PPM [5] je eden izmed serije formatov za shranjevanje bitnih slik. Datoteka je sestavljena iz dveh delov, glave in podatkov. V glavi (slika 2.1) so vsaj tri vrstice s specifikacijami. V prvi vrstici je PPM identifikator, v drugi vrstici je zapis o širini in višini slike, v tretji pa najvišja vrednost barvne komponente za piksel. Poleg obveznih vrstic je dovoljen komentar z začetnim znakom #. Format podatkov slike je določen z identifikatorjem 'P3' z ASCII kodiranjem.

```
P3
2 2
# komentar
255
```

Slika 2.1: Primer glave.

Sledi podatkovni del, ki določa sliko. Vsak piksel je določen s tremi 8-bitnimi vrednostmi in skupaj določajo barvo z barvnega modela RGB kjer je R rdeča, G zelena, B pa modra barvna komponenta (slika 2.2). V našem primeru je smiselna uporaba le črne in bele barve, vendar je dovoljena uporaba tudi barvnih 24-bitnih slik. Program, ki je bil razvit za delo z vodnim tiskalnikom samodejno pretvori barvno sliko v črno-belo na podlagi vsote barvnih komponent.

```
P3
2 2
# komentar
255
0 255 0 0 0 255
255 255 255 0 0 0
```



Slika 2.2: Primer slike v PPM formatu.

Zavedati se moramo, da je PPM format neučinkovit. Vsebuje veliko informacij, ki jih človeško oko ne zazna, poleg tega pa razen osnovnih barv vsebuje bore malo informacij o sliki. Po drugi strani pa je zelo enostaven za branje prav tako kot tudi izdelava programov za manipulacijo, kar je bistveno za naš tiskalnik.

## 2.2 Program za risanje

Napisal sem program za risanje, ki pretvori barvno sliko v datoteko enic in ničel, ki je razumljiva elektroni. Programska maska programa za risanje je v splošnem sestavljena iz štirih delov; mreže, ki predstavlja risalno površino, pogled izhodne datoteke enic in ničel, skupine gumbov za delo ter vrstic stanja. Program omogoča odpiranje datotek PPM in tekstovnih datotek TXT ter pretvorbo slike v datoteko enic in ničel.

Ob odprtju datoteke PPM ali TXT program prebere vrednosti in jih interpretira v polju. Za vsak piksel program na osnovi vsote treh komponent določi barvo celice, ki jo obarva (Slika 2.3).



Širina slike, v našem primeru 80 pikslov

Slika 2.3: Prikaz slike v polju.

Program je namenjen prikazovanju slik resolucije  $80 * Y$ , kjer je 80 širina slike, določena s številom šob in  $Y$  poljubna višina slike. Širina posameznega kvadratka je proporcionalno določena s širino šobe, višina pa z dolžino kapljice. Za prikaz je privzeta idealna kapljica v obliki kvadratka.

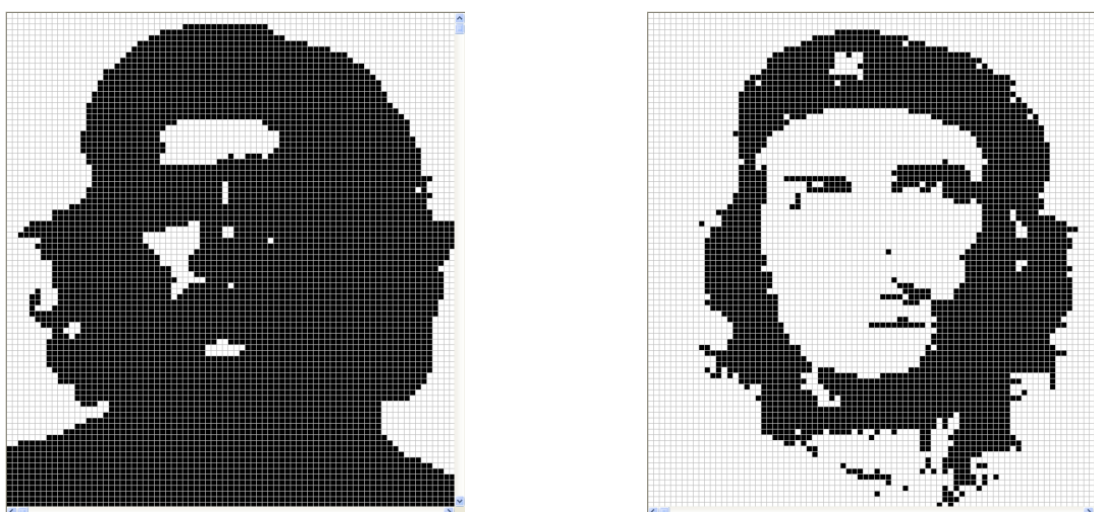
Program poleg prikazovanja že narisanih slik omogoča tudi osnovno urejanje. V sliki lahko rišemo ali brišemo posamezne kvadratke. Končni rezultat vodnega printerja je vodna slika, torej lahko le-ta zavzame dve vrednosti, voda ali praznina. Ker pa je večina slik v barvah, jih

je potrebno pretvoriti v binarno sliko, za kar poskrbi aplikacija. Na voljo je torej 1 bit za posamezen piksel.



Slika 2.4: Pretvorba v binarno sliko.

Pri pretvorbi v binarno sliko (Slika 2.4) se izgubijo vse informacije o sencah in barvah. V primeru je uporabljena 24-bitna črna bela slika, ki se pri pretvorbi precej spremeni. Še bolj opazna je sprememba pri pretvorbi barvnih slik, saj rumena barva pomeni belo. Rešitev problema in uporabnikovo odločitev kaj se bo prikazalo in kaj ne je določitev meje med črno in belo barvo. S tem so po eni strani izločene nepomembne barvne sence, po drugi strani pa so z ustrezno izbiro meje prikazane vse zelene sence oziroma detajli, ki bi sicer ostali nevidni (Slika 2.5).



Slika 2.5: Vpliv pomika meje med črno in belo barvo na sliki.



# Poglavje 3

## Vodni tiskalnik

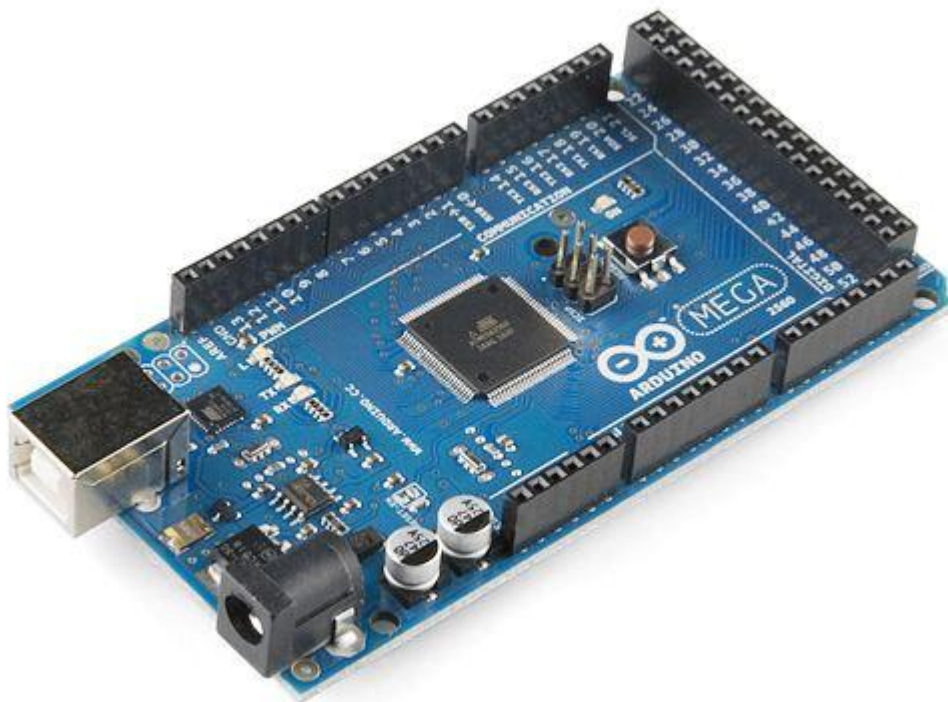
### 3.1 Vezje

Za krmiljenje vodnega tiskalnika je uporabljeno razvojno orodje Arduino Mega2560. Kljub Arduinovem visokem številu izhodov je le-teh za rešitev omenjenega problema premalo in je za razširitev uporabljenih 10 pomikalnih registrov, s katerimi dobimo dodatnih 80 izhodov. Ker so elektromagneti napajani z 220V izmenične napetosti, na izhodu pomikalnih registrov pa imamo 5V enosmerne napetosti, je potrebno izdelati vmesno vezje, ki uporabi 5V za proženje 220V.

#### 3.1.1 Arduino Mega2560

Arduino je odprtokodna platforma namenjena izdelovanju elektronskih prototipov. Zaradi svoje fleksibilnosti je zelo priljubljena med začetniki vendar s svojim širokim naborom knjižnic omogoča izdelavo kompleksnih elektronskih sistemov.

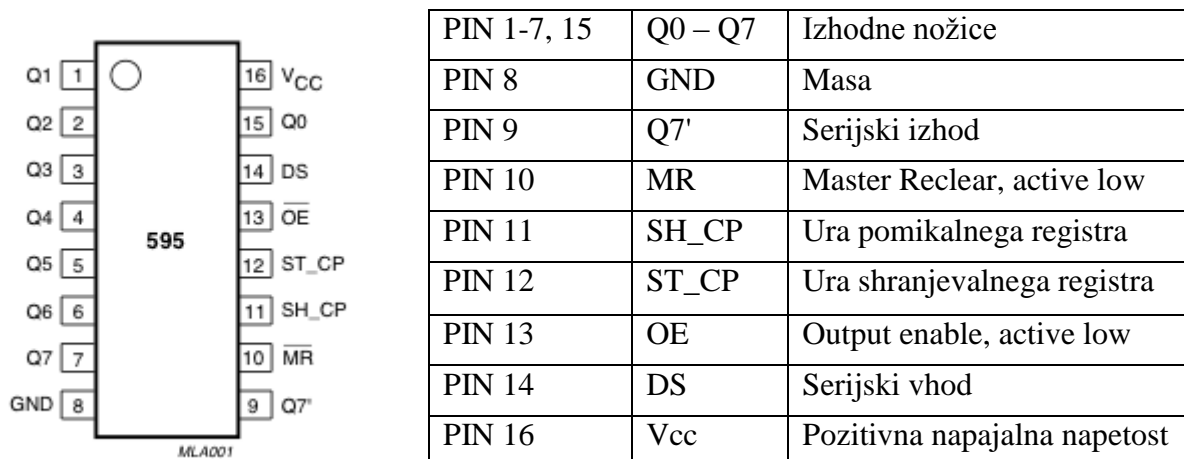
Razvojno orodje Arduino Mega2560 [6] (Slika 3.1) temelji na Atmelovem mikrokrmilniku Atmega2560. Ima 54 digitalnih vhodov/izhodov ter 16 analognih. Arduino je programljiv preko vtiča USB in omogoča napajanje preko baterije. Nanj je mogoče priklopiti različne naprave kot sta čitalec kartic in vmesnik Bluetooth.



Slika 3.1: Razvojno orodje Arduino Mega 2560.

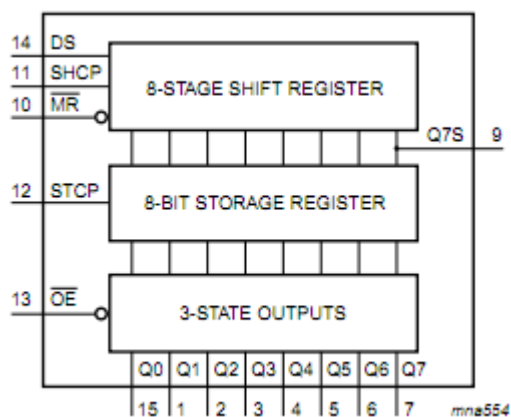
### 3.1.2 Pomikalni register

Pri priključevanju velikega števila naprav se srečamo z omejitvijo priključkov, ki jih ima razvojno orodje Arduino. V ta namen lahko število izhodov razširimo s pomočjo pomikalnih registrov [7]. Register je spominska enota za hranjenje več bitov. Pomikalni register je sestavljen iz več kaskadno vezanih pomnilnih celic, pri čemer se stanje celic v taktu ure prenaša iz ene celice v drugo. Glede na prenos podatkov ločimo več osnovnih vrst pomikalnih registrov, v nalogi so uporabljeni 8-bitni SIPO (zaporedni vhod – vzporedni izhod) registri 74HC595 [8]. Po specifikacijah je frekvenca pomikanja podatkov 100MHz, kar je za naš primer dovolj.



Slika 3.2: Pomikalni register 74HC595 s seznamom nožic.

Podatki se pomikajo ob pozitivnem prehodu Ure pomikalnega registra (SH\_CP) (Slika 3.2) in se ob pozitivnem prehodu Ure shranjevalnega registra (ST\_CP) prenesejo v shranjevalni register in posledično na izhode.



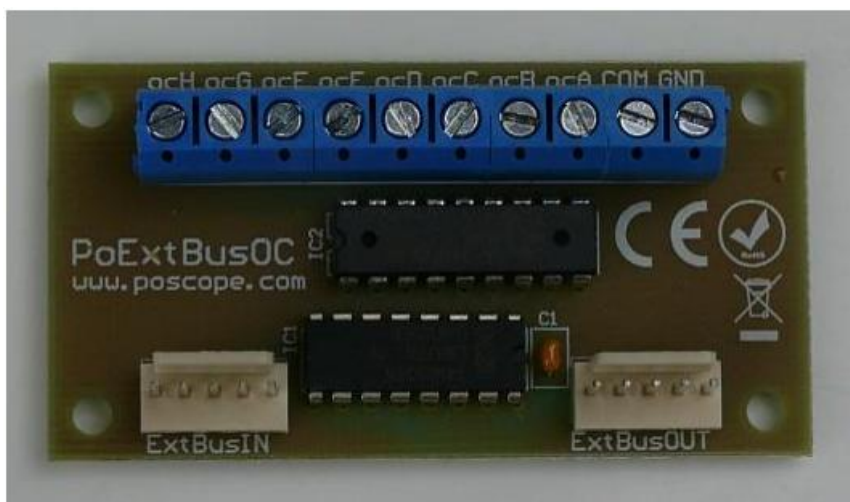
Slika 3.3: Zgradba pomikalnega registra.

Prenos poteka serijsko sinhrono, kar pomeni, da s spreminjanjem nivoja signalov DS pošljamo bajt v register bit po bit. Ko je celoten bajt prenesen v register, se s prehodom signala ST\_CP iz HIGH v LOW vrednost paralelno prenese na izhodne nožice (Slika 3.3).

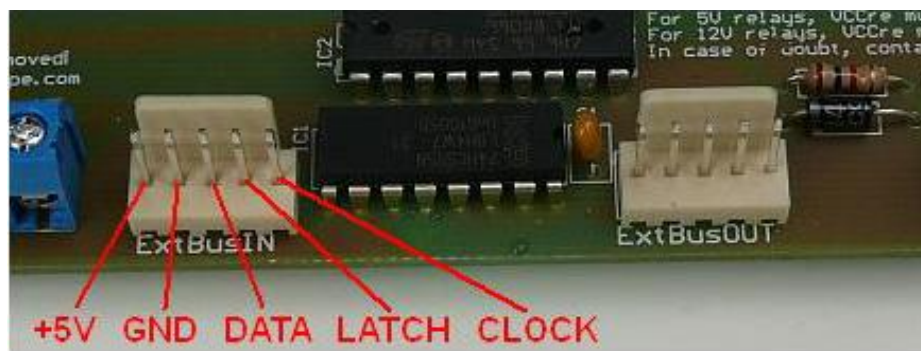


Pomikalni registri so namenjeni krmiljenju majhnega vezja (opisano v nadaljevanju) (Slika 3.8), ki vklaplja elektromagnete. Vezje za vklop potrebuje na vhodu vsaj 35mA toka, vendar je na posameznih kanalih izhoda pomikalnih registrov prisotnih le 20mA. Za zagotovitev dovolj toka je na izhod pomikalnega registra vezan ojačevalnik ULN2803A [9], ki lahko krmili tokove do 500mA na kanal.

Uporabljenih je deset vezij PoExtBusOC [10] (Slika 3.5), ki že vsebujejo pomikalne registre 74HC595 in ojačevalnike ULN2803A.



Slika 3.5: Vezje PoExtBusOC s pomikalnim registrom in ojačevalnikom.



Slika 3.6: Priklučitev vezja.

Za kaskadno vezavo je potrebno povezati istoležne nožice izhoda enega vezja in vhoda naslednjega vezja v verigi (Slika 3.6).

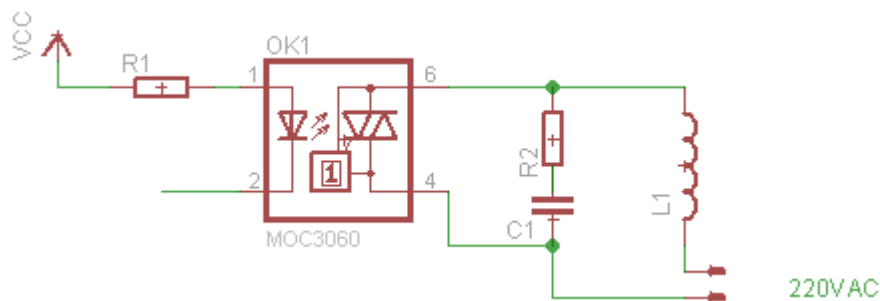
### 3.1.3 Optospojnik

Elektromagneti so napajani z 220V izmenične napetosti, na izhodu ojačevalnikov pa imamo 5V enosmerne napetosti. Potrebno je izdelati vmesno vezje, ki uporabi 5V za proženje 220V. V ta namen je uporabljen optospojnik MOC3060 [11] (Slika 3.7).



Slika 3.7: Optospojnik MOC3060.

Ker napetost in tok nista v fazi, se je v praksi izkazalo, da je na izhod optospojnika potrebno vezati RC blažilnik (RC snubber [12]), ki zaduši hitre dvige napetosti čez triak in preprečuje napačen vklop. Problem se pojavlja pri nelinearnih bremenih kot so motorji ali elektromagneti, kar je primer v našem tiskalniku.



Slika 3.8: Shema vezja za vklop elektromagneta.

### 3.1.4 Program za Arduino

Program za Arduino skrbi za prenos bitov oz. bajtov slike enic in ničel na pomikalne registre. Program v zanki bere vrstice in jo bit po bit z ukazi knjižnice ShiftOut [13] pošilja na nožico DS. Pomikanje bitov poteka ob prehajanju signala SH\_CP iz nizkega v visoko stanje, pojavitev vseh bitov na izhodne nožice ojačevalnika pa se zgodi ob pozitivnem prehodu signala SH\_CP.

#### Nastavitve izhodov Arduina za krmiljenje pomikalnih registrov

```
...  
// Nožica priključena na ST_CP  
const int latchPin = 8;  
  
// Nožica priključena na SH_CP  
const int clockPin = 12;  
  
// Nožica priključena na DS  
const int dataPin = 11;  
...
```

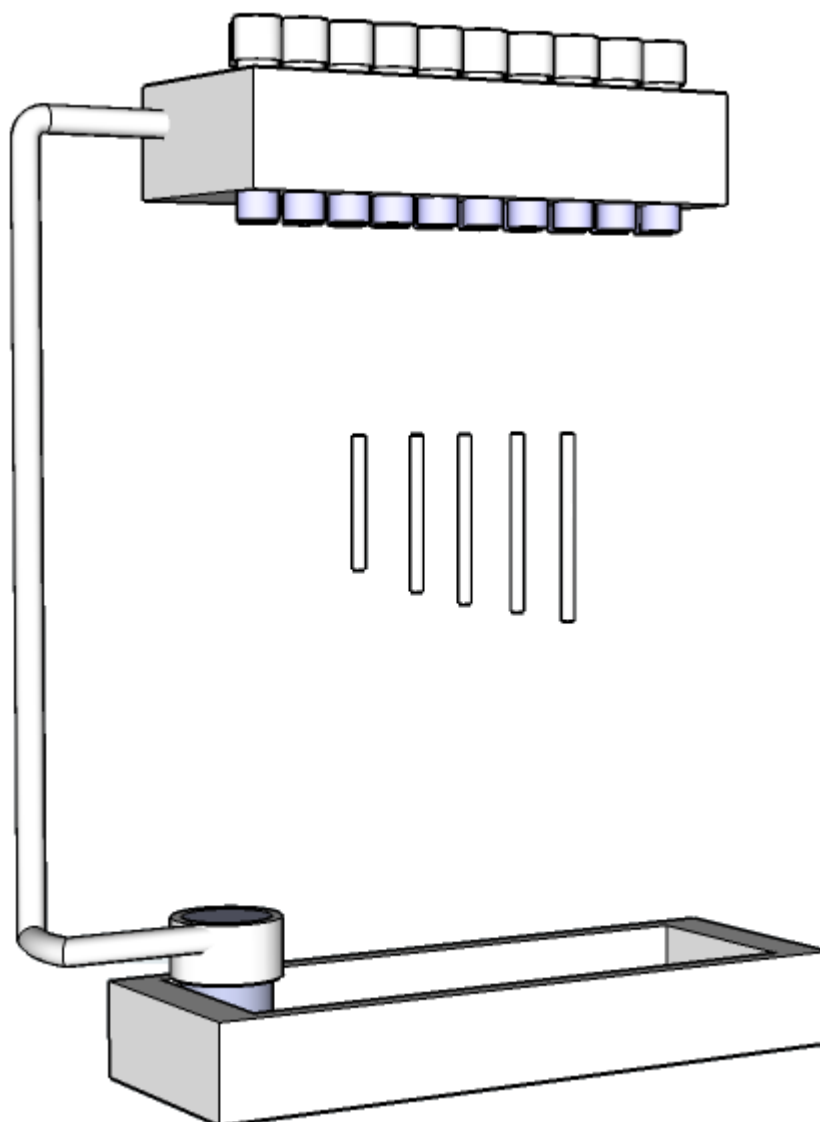
#### Primer programa uporabe pomikalnega registra

```
...  
void loop() {  
  for (int i = 0; i < 80; i++) {  
    digitalWrite(latchPin, 0);  
    //V času pošiljanja je latchPin v nizkem stanju  
    for (int j = 0; j < 80; j++){  
      shiftOut(dataPin, clockPin, tabela[i][j]);  
      //ob prehodu signala clockPin v visoko stanje se biti pomaknejo  
    }  
    digitalWrite(latchPin, 1);  
    delay(pavza);  
  }  
}  
...
```

Slika 3.9: Nastavitve izhodov in primer uporabe.

### 3.2 Vodni segment

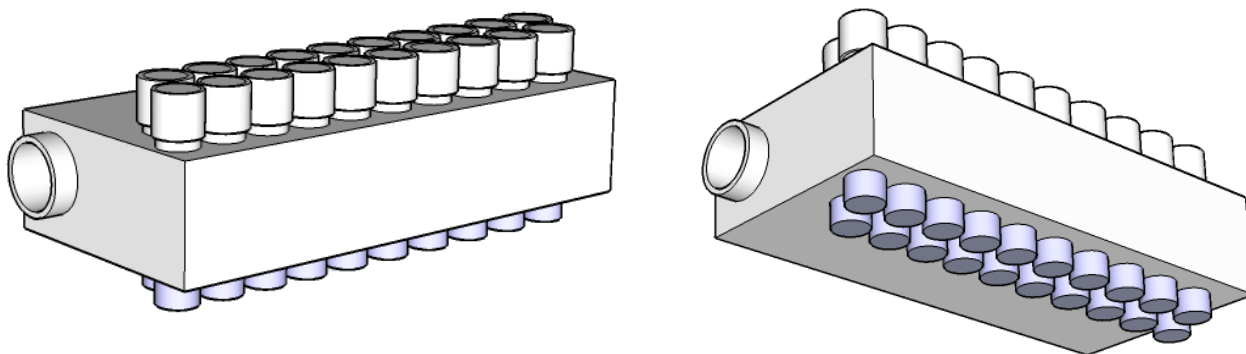
Vodni segment (Slika 3.10) je del tiskalnika, ki skrbi za prikazovanje slik v obliki vodnega slapa. Njegova naloga je odpiranje ventilov in skrb za pravilno distribucijo vode po rezervoarjih. Sestavni deli vodnega segmenta so zapirala šob (Slika 3.12), rezervoar v katerem so nameščene šobe (Slika 3.11) z elektromagneti, zbiralni rezervoar s potopno črpalko (Slika 3.13), ki skrbi za prečrpavanje vode ter cevi.



Slika 3.10: Vodni segment.

### 3.2.1 Rezervoar s šobami

Vodni tiskalnik je oblikovan tako, da spušča vodo z vrha, pri čemer je višina odvisna od lastnosti šob. Šobe so nameščene na dnu rezervoarja in skrbijo za pravilno razporeditev vode v vodni slap. Običajen vodni slap dobimo v primeru, ko so vse šobe odprte in voda prosto pada iz rezervoarja, v nasprotnem primeru je voda ujeta. Vidljivost slike je odvisna od lomnega količnika vpadne svetlobe na padajočo vodo in je tako orientacija šob povsem odvisna od fizične postavitve tiskalnika. Na vrhu rezervoarja je priklop za cev, po kateri se voda prečrpa iz spodnjega zbiralnika. Tako se voda reciklira in je s tem strošek polnjenja manjši.



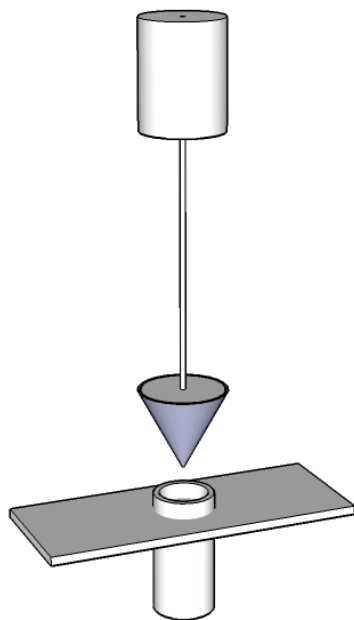
Slika 3.11: Rezervoar s šobami. Razporejene so v dve vrsti s 40 šobami v vsaki vrsti.

Šobe v rezervoarju morajo biti dovolj skupaj, da je slika lepo vidna, po drugi strani pa premajhen razmak pomeni zlivanje sosednjih curkov. Preveliki razdalji med šobami se lahko ognemo tako, da šobe postavimo v dve vrsti, ki sta med seboj zamaknjeni. S tako postavitvijo so šobe dovolj narazen, obenem pa se prikaz bistveno ne spremeni.

Šobe morajo biti dovolj dolge, saj voda potrebuje nekaj časa, da se oblikuje v curek. Poleg dolžine morajo biti šobe primerno ozke, voda v njej zadrži in se ne izlije. Problem lahko analogno prenesemo na slamico z enim zaprtim koncem in na kozarec. Zaradi površinske napetosti se iz slamice voda ne izlije, pri kozarcu pa je teža vode prevelika in se voda izlije. V primeru, ko je največji dovoljen premer šobe še zmeraj vizualno premajhen, lahko šobe združujemo.

### 3.2.2 Zapiralo šobe

Zapiralo šobe je zadnji člen v verigi in skrbi za izstavitve vode. Zapiralo sestoji iz gumijastega stožca, pritrjenega na kovinsko palico. Palica deluje kot feromagnetno jedro elektromagneta. Ko je elektromagnet neaktiven, gumijasti stožec leži na odprtini šobe in preprečuje vodi, da se izlije. V trenutku pojavitve električne napetosti, pa elektromagnet potegne palico in šoba se sprosti.

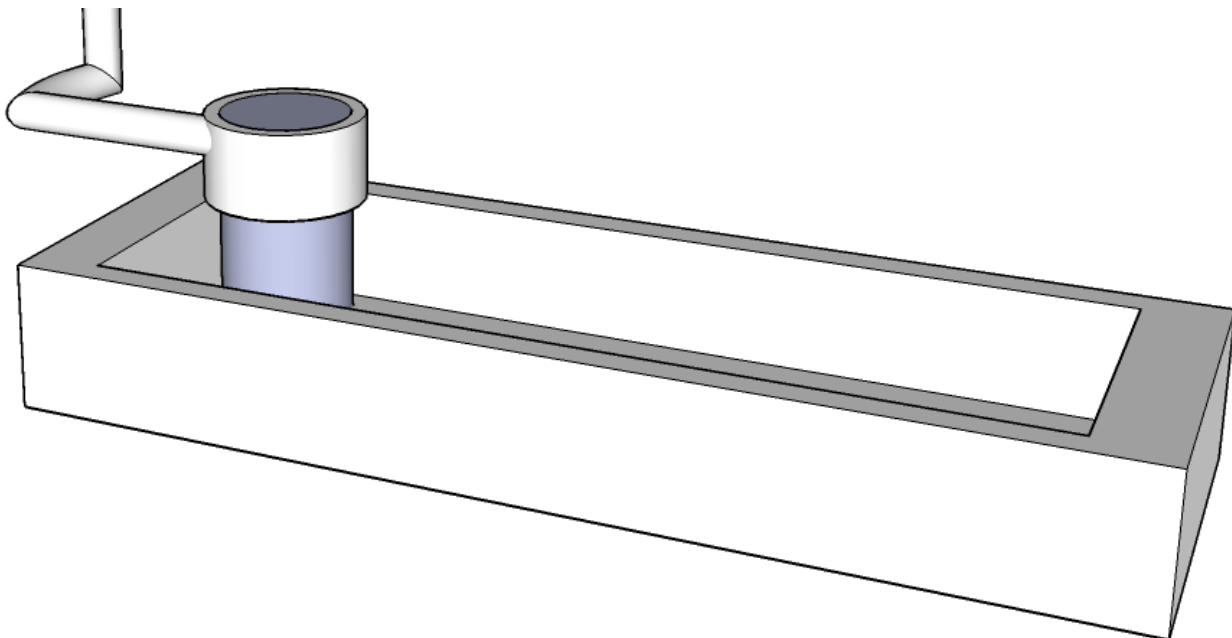


Slika 3.12: Zapiralo šobe.

V primeru, ko so vsi elektromagneti aktivni, so vse šobe odprte in voda teče v slapu. Elektronika skrbi za krmiljenje elektromagnetov in tako kot rezultat dobimo vodno sliko. Šobe lahko razporedimo v več vrst, pri čemer posamezen elektromagnet odpira celotno kolono. S povečevanjem vrst vplivamo na odboj svetlobe in s tem vplivamo na vidljivost. Šobe lahko združujemo in pridobimo na širini, pri čemer resolucija izrisa ostaja enaka. V našem primeru združevanja ni in vsak elektromagnet odpira le eno šobo.

### 3.2.3 Zbiralni rezervoar s potopno črpalko

Vodni tiskalnik je zasnovan kot zaprti sistem. Porabljena voda kroži, kar je zaželeno v okolju kjer dotok vode ni stalno zagotovljen. V ta namen je na dnu postavljen zbiralnik, ki nabrano vodo prečrpava nazaj v rezervoar s šobami.



Slika 3.13: Zbiralnik s potopno črpalko.

Voda se preko cevi prečrpava s pomočjo potopne črpalke (Slika 3.14), ki je postavljena ob rob rezervoarja. Uporabljena črpalka vsebuje senzor nivoja vode, ki je uporaben le pri velikih rezervoarjih, kjer nivo zelo niha. Pri manjših rezervoarjih je potrebno zagotoviti natančnejši senzor, ki je lahko postavljen v rezervoarju s šobami.

V našem primeru senzor ni uporabljen. Zaradi težnje h konstantnemu tlaku v rezervoarju s šobami, je črpalka stalno v delovanju. Rezervoar je vodotesen in dobava vode je večja kot je izpust skozi šobe. Zaradi narave potopne črpalke se ob zapolnitvi rezervoarja voda ne dobavlja več in tako je v rezervoarju tlak konstanten.

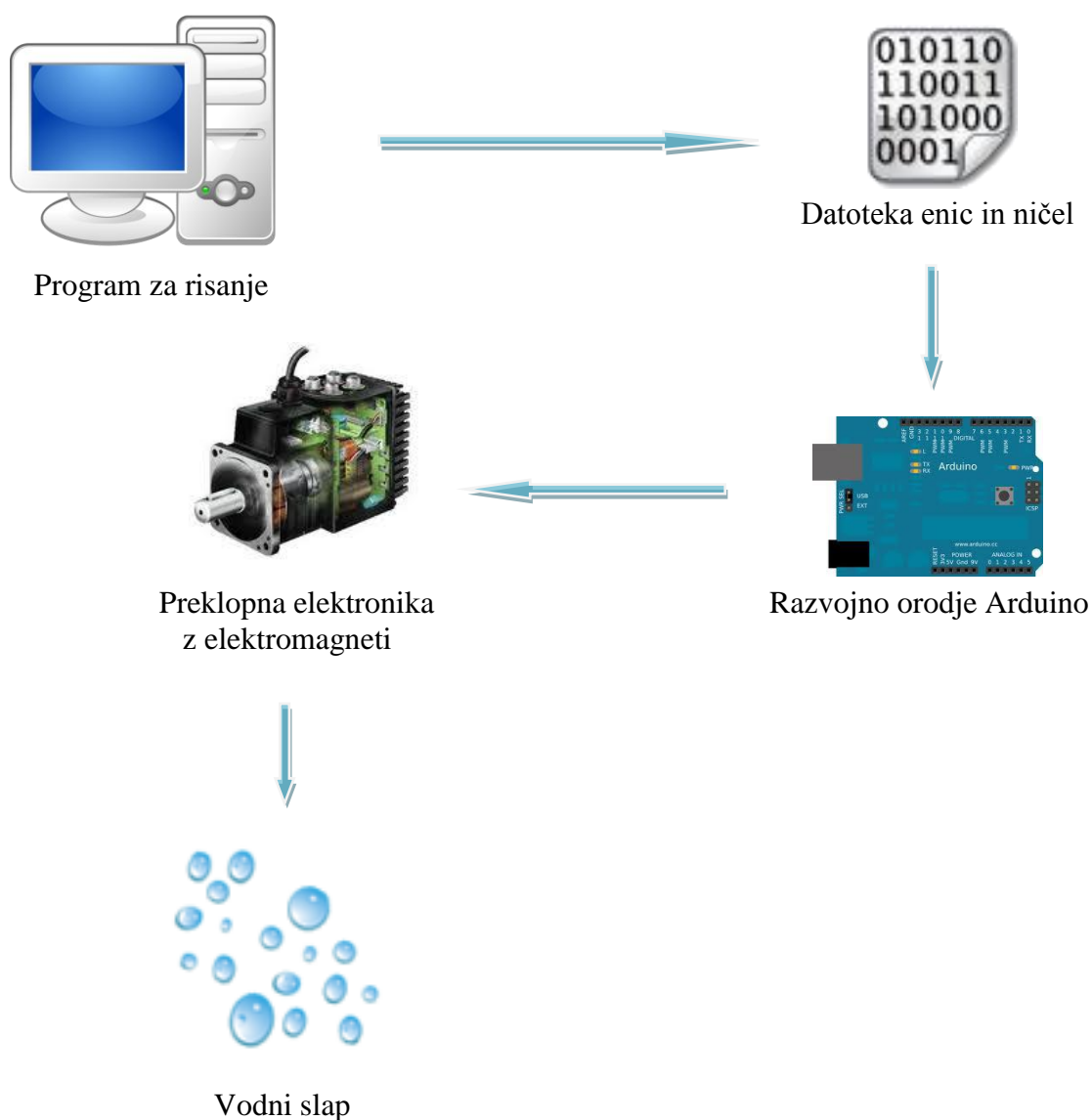


Slika 3.14: Potopna črpalka.

Potopna črpalka je izbrana z Alkovega programa potopnih črpalk z 10.000 litri pretoka na uro[14]. Takšen pretok je zadosten, vendar je potrebno cev razširiti. Z širitvijo cevi reguliramo tlak, ki ga črpalka ustvari, saj lahko pri majhnem rezervoarju s šobami nezaželeno vpliva na razporejanje vode. Za ta projekt bi lahko bila izbrana katera koli vrsta črpalke, lahko bi bila tudi postavljena ob rezervoar s šobami, vendar se je razmerje vodni pretok/cena izkazalo pri omenjeni črpalki kot najugodnejše.

### 3.3 Delovanje vodnega tiskalnika

Za delovanje vodnega tiskalnika (Slika 3.15) je potrebno s programom za risanje ustvariti sliko in jo pretvoriti v datoteko enic in ničel. Vsebino datoteke prenesemo v tabelo Arduinovega programa in ga naložimo. Pred zagonom je potrebno vodo natočiti v zbiralni rezervoar in pognati črpalko. Ko je voda v rezervoarju s šobami, lahko program zaženemo.



Slika 3.15: Delovanje vodnega tiskalnika.

## Poglavje 4

# Sklepne ugotovitve

V okviru diplomske naloge sem želel narediti izdelek, ki bi na tržišču turizma predstavljal popestritev zunanjega izgleda poslopij. Izdelal sem vodni tiskalnik, ki je zmožen prikazovati slike, vzorce in napise. Vodni tiskalnik je širine 130cm s skupno osemdesetimi šobami, razporejenimi v dve vrsti. Program za risanje je izdelan v programskem jeziku Object Pascal, za upravljanje z elektromagnetnimi ventili pa skrbi Arduino.

Program za risanje pri razvoju ni predstavljal večjih problemov, prav tako ne elektronika. Vsi elektronski elementi so v času testiranja delovali brez posebnosti. Zaradi možnih obremenitev so bili preventivno zamenjani dovodni kabli. Rezervoar s šobami je narejen iz pleksi stekla širine 4mm, kar se je izkazalo za napačno izbiro. Plošče bi idealno morale biti iz trdega materiala, ki se ne bi upogibal. Okvirji rezervoarja so izdelani iz aluminijastih profilov, ki držijo rezervoar skupaj. Zapirala šob sestavljajo elementi, ki temu niso namenjeni in tak je tudi rezultat. Nekaj materiala, ki sem ga nabavljal, je bilo očitno iz različnih serij in kljub isti ceni in kodi so bila odstopanja prevelika. Zaradi teh razlik so bile med izdelavo rezervoarja potrebne spremembe načrtov. Pri rezervoarju sem naletel še na problem tesnenja, saj elektromagnetov nisem uspel neprodušno namestiti. Zaradi tega je bil dovod vode ročno krmiljen. To je, vodo sem dotočil kadar je bilo to potrebno.

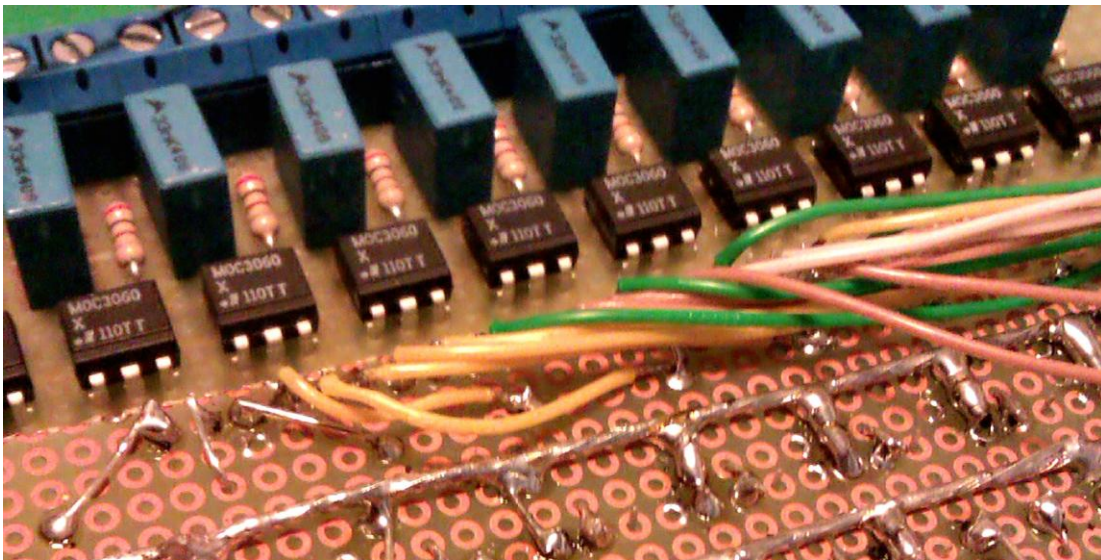
Ko sem vodni tiskalnik zagnal prvič, se je izrisala samo polovica slike. Ugotovil sem, da se zaradi Arduinove omejitve pomnilnika vanj ne naloži celotna tabela enic in ničel. Program se prevede, prenese pa se samo del tabele. Tabelo celih števil sem nadomestil s tabelo bajtov in problem je bil navidezno rešen. Elegantnejša rešitev bi bila z uporabo SD modula, ki bi s spominske kartice bral cele bajte datoteke enic in ničel in jih ciklično interpretiral. Pri tako shranjeni sliki se poraba prostora bistveno zmanjša.

Ne glede na omenjene probleme, na katere sem med izdelavo naletel, so pričakovanja dosežena. Privzeti je potrebno, da se do prvega zagona ni vedelo kakšen bo rezultat in ga je bilo težko napovedati.

Z diplomskim delom sem tako poglobil znanje o mikrokontrolerih, načrtovanju in izdelavi elektronskih vezij. Predvsem sem pa ugotovil, da je potrebno imeti dobro testno okolje, če želimo vsaj približno predvideti rezultat dela oz. končnih izdelkov. Vodni tiskalnik bo deležen še številnih sprememb in dodelav, predvsem pa bodo upoštevane vse ideje, ki so se porodile med izdelavo in jih je bilo trenutno težko realizirati.

# Dodatek A

## A.1 Slike vodnega tiskalnika



Slika A.1: Elektronika.



Slika A.2: Vodni tiskalnik (pogled na šobe).



Slika A.3: Vodni curki.



Slika A.4: Primer izrisanih vzorcev 1.



Slika A.5: Primer izrisanih vzorcev 2.



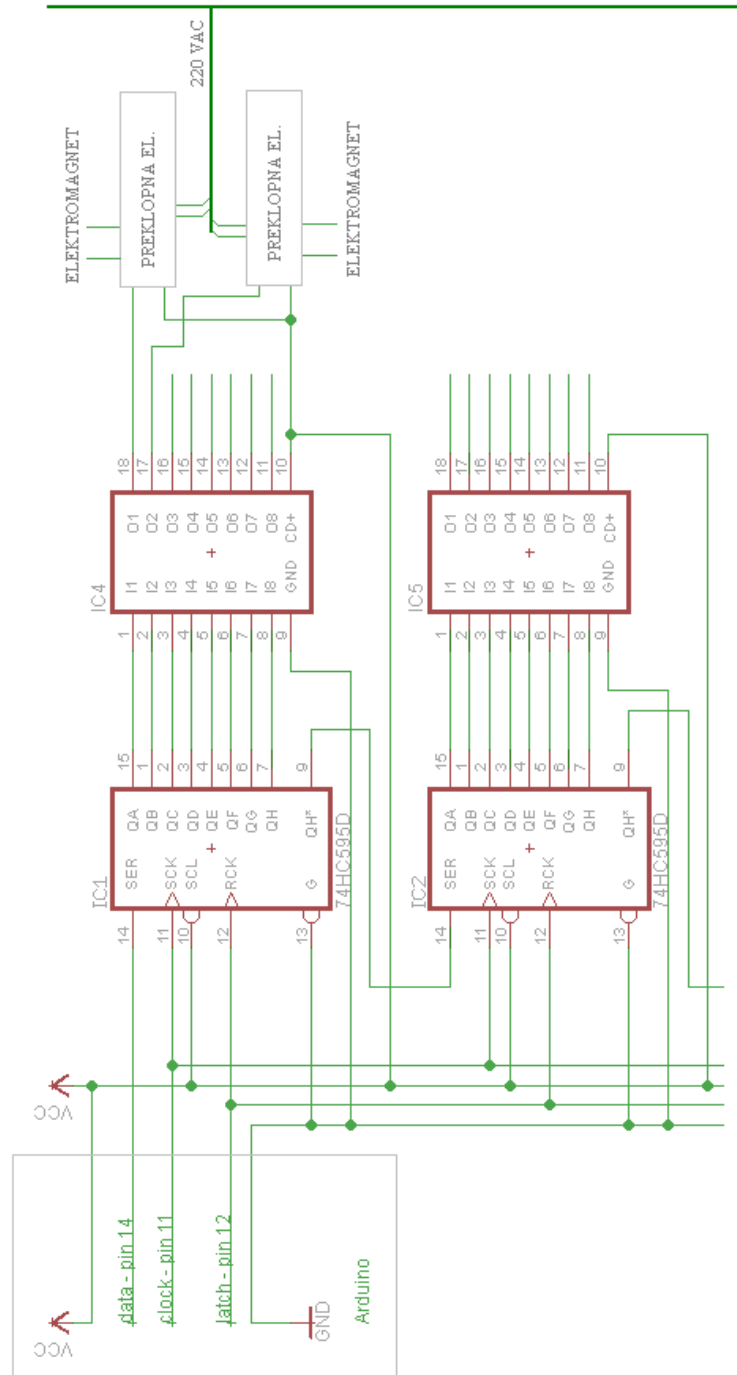
Slika A.6: Primer izrisanih vzorcev 3.

## A.2 Seznam slik

2.1	Primer glave	5
2.2	Primer slike v PPM formatu	5
2.3	Prikaz slike v polju	6
2.4	Pretvorba v binarno sliko	7
2.5	Pomik meje med črno in belo barvo na sliki	7
2.6	Izhodna datoteka enic in ničel	8
3.1	Razvojno orodje Arduino Mega 2560	10
3.2	Pomikalni register 74HC595 s seznamom nožic	11
3.3	Zgradba pomikalnega registra	11
3.4	Pomikalni registri z ojačevalniki ULN2803A	12
3.5	Vezje PoExtBusOC s pomikalnim registrom in ojačevalnikom	13
3.6	Priključitev vezja	13
3.7	Optospojnik MOC3060	14
3.8	Shema vezja za vklop elektromagneta	14
3.9	Nastavitve izhodov in primer uporabe	15
3.10	Vodni segment	16
3.11	Rezervoar s šobami	17
3.12	Zapiralo šobe	18
3.13	Zbiralnik s potopno črpalko	19
3.14	Potopna črpalka	20
3.15	Delovanje vodnega tiskalnika	21
A.1	Elektronika	24
A.2	Vodni tiskalnik (pogled na šobe)	25
A.3	Vodni curki	25
A.4	Primer izrisanih vzorcev 1	26
A.5	Primer izrisanih vzorcev 2	26
A.6	Primer izrisanih vzorcev 3	27
B.1	Shema elektronike	29

# Dodatek B

## B.1 Shema elektronike



Slika B.1 Shema elektronike.

# Dodatek C

## C.1 Priložen CD

- Izvorna koda programa za risanje v Delphiju,
- Izvorna koda programa za razvojno orodje Arduino,
- Predstavitveni video končnega izdelka s primeri uporabe.

# Viri in literatura

- [1] (2009) Nakupovalno središče Fukuoka, Japonska. Dostopno na: <http://www.youtube.com/watch?v=Es4kfg0Gxs>
- [2] (2006) Reklama za Jeep na avto sejmi. Dostopno na: <http://www.youtube.com/watch?v=Z2LUz2WVcek>
- [3] (2010) Uradna stran Pevnick Design, inc. Dostopno na: <http://www.pevnickdesign.com/>
- [4] (2011) Embarcadero Delphi, wikipedia. Dostopno na: [http://sl.wikipedia.org/wiki/Embarcadero\\_Delphi](http://sl.wikipedia.org/wiki/Embarcadero_Delphi)
- [5] (2003) PPM, sourceforge. Dostopno na: <http://netpbm.sourceforge.net/doc/ppm.html>
- [6] (2010) Arduino board Mega2560. Dostopno na: <http://arduino.cc/en/Main/ArduinoBoardMega2560>
- [7] (2011) Specifikacije pomikalnega registra 74HC595. Dostopno na: [http://www.nxp.com/documents/data\\_sheet/74HC\\_HCT595.pdf](http://www.nxp.com/documents/data_sheet/74HC_HCT595.pdf)
- [8] (2011) Splošno o pomikalnih registrih, wikipedia. Dostopno na: [http://en.wikipedia.org/wiki/Shift\\_register](http://en.wikipedia.org/wiki/Shift_register)
- [9] (2006) Specifikacije za gonilnik ULN 2803a. Dostopno na: <http://focus.ti.com/lit/ds/symlink/uln2803a.pdf>
- [10] (2011) Predstavitev ploščice PoExtBusOC. Dostopno na: <http://www.poscope.com/product.php?pid=26>
- [11] (2004) Specifikacije za optospojnik MOC3060. Dostopno na: <http://www.optoinc.com/datasheets/MOC3060.pdf>
- [12] (2003) Shema RC blažilnika. Dostopno na: [http://www.soloelectronica.net/triac/AN\\_3003.PDF](http://www.soloelectronica.net/triac/AN_3003.PDF)

- [13] (2006) Shift Out Library. Dostopno na:  
<http://www.arduino.cc/en/Tutorial/ShiftOut>
- [14] (2010) Alkov program potopnih črpalk. Dostopno na:  
<http://www4.al-ko.de/geschaeftsfelder/garten-hobby/produkte/wassertechnik/combi-universaltauchpumpen/combi-universaltauchpumpe-twin-10000-combi.html>