



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

Tim Kosec

Zatemnilnik za LED luči

Kazalo

1 UVOD	3
2 ZASNOVA	4
2.1 TEORIJA	4
2.2 BLOKOVNI DIAGRAM	5
3 ALTIVM DESIGNER	6
3.1 SHEMA	6
3.2 TISKANINA	8
3.3 SEZNAM KOMPONENT	10
3.4 GERBER	11
4 SESTAVA ELEKTRONIKE	11
4.1 SESTAVA	11
4.2 PROGRAMIRANJE	13
4.3 TESTIRANJE	14
5 OPIS DELOVANJA	15
6 ZAKLJUČEK	15
7 PROGRAM	16

Kazalo slik

SLIKA 1 BLOKOVNI DIAGRAM VEZJA	5
SLIKA 2 SHEMA VEZJA Z IZBRANIMI VREDNOSTMI ELEMENTOV	7
SLIKA 3 ZAVIHEK S PRAVILI DIZAJNIRANJA TISKANINE	8
SLIKA 4 TISKANO VEZJE V UREJEVALNIKU	9
SLIKA 5 2D POGLED TISKANEGA VEZJA	9
SLIKA 6 3D POGLED TISKANEGA VEZJA	10
SLIKA 7 SEZNAM KOMPONENT (BOM)	10
SLIKA 8 TISKANO VEZJE BREZ ELEMENTOV	11
SLIKA 9 KONČANO TISKANO VEZJE	12
SLIKA 10 STRANSKI POGLED KONČANEGA TISKANEGA VEZJA	12
SLIKA 11 POGLED KONČANEGA TISKANEGA VEZJA OD SPODAJ	12
SLIKA 12 PRIKLJUČITEV PROCESORJA ZA NALAGANJE PROGRAMA	13
SLIKA 13 SIGNAL NA IZHODU VEZJA	14
SLIKA 14 PODATKOVNI SIGNAL SIPO REGISTRA	14

1 Uvod

V poročilu je opisan celoten postopek izdelave elektronike od ideje do izdelka. Vsebuje izdelavo tiskanega vezja za zatemnjevanje enosmernih LED luči.

V prvem delu je opisana zasnova v teoriji. Tukaj je razložen postopek, kako priti od problema do ideje za projekt ter kako ga načrtovati. Nato sledi izdelava sheme, kjer izberemo prave komponente ter izrišemo elektroniko v CAD programu. Pri tem se srečamo z težavami razpoložljivosti komponent ter spoznamo programsko okolje Altium. Na koncu še sestavimo tiskano vezje in ga testiramo. Poročilu je dodan tudi program za mikroprocesor.

Celoten postopek je dokaj preprost, tako da se lahko kdor koli odloči za izdelavo tega projekta če ima vsaj malo zanimanja. Zajema obsežno znanje ter je zabavno za izdelavo in uporabno v praksi.

Poročilo vsebuje:

- idejni potek realiziranja elektronike
- blokovni diagram
- izbira elementov
- zasnova sheme
- nastavitev omejitev proizvajalca
- izris PCB
- sestava seznama elementov (BOM)
- generiranje gerber datoteke
- spajkanje PCB
- programiranje mikroprocesorja
- testiranje elektronike
- program (C++)

Programska oprema:

- Altium designer
- Arduino IDE

Dobavitelji:

- Farnell
- JLC pcb

2 Zasnova

2.1 Teorija

Za moj projekt sem si izbral elektroniko za zatemnjevanje enosmernih LED luči. Razlog za izbiro je ta, da sem potreboval rešitev v sobi. Težava je bila v osvetljenosti delovnega prostora ter metanje sence. Potreba je bila rešitev. Za odpravo te težave sem uporabil 12 V LED trak ter ga namestil na optimalne površine. Trak je težavo odpravil vendar v prostoru ni vedno potrebna maksimalna svetlost, Tukaj nastopi zatemnilnik. Zatemnilnik je naprava, ki analogno ali digitalno zmanjšuje napetost, kar posledično zmanjša svetilnost svetlobnega vira. V mojem projektu je uporabljen digitalni princip, ki je podrobneje opisan v nadaljevanju.

Ker sem potreboval cenovno ugodno, majhno ter primerno rešitev, sem se odločil elektroniko izdelati sam. Prvi korak je bil teoretični opis delovanja. Zastavil sem si cilj, da ta elektronika potrebuje vhodni vmesnik, prikazovalnik ter gonilnik za izhod. Za vhod je bilo najbolj smiselno izbrati preprosto napravo, s pomočjo katere je mogoče nastavljati svetilnost hitro, natančno ter je cenovno ugoden. Na podlagi teh zahtev sem izbral potenciometer oziroma nastavljiv upor. Pri izbiri prikazovalnika so bile zahteve podobne, minimalističen izgled ter cenovno ugodna rešitev. Izbral sem LED niz, ki vsebuje 10 rdečih LED sijalk. Izbrati je bilo potrebno še način, kako spreminjati napetost na izhodu. Najočitnejša rešitev je bila, pulzno širinska modulacija (PWM). Prednost tega načina so visoka učinkovitost, kompaktnost ter preprosta izvedba, saj je potreben le en tranzistor.

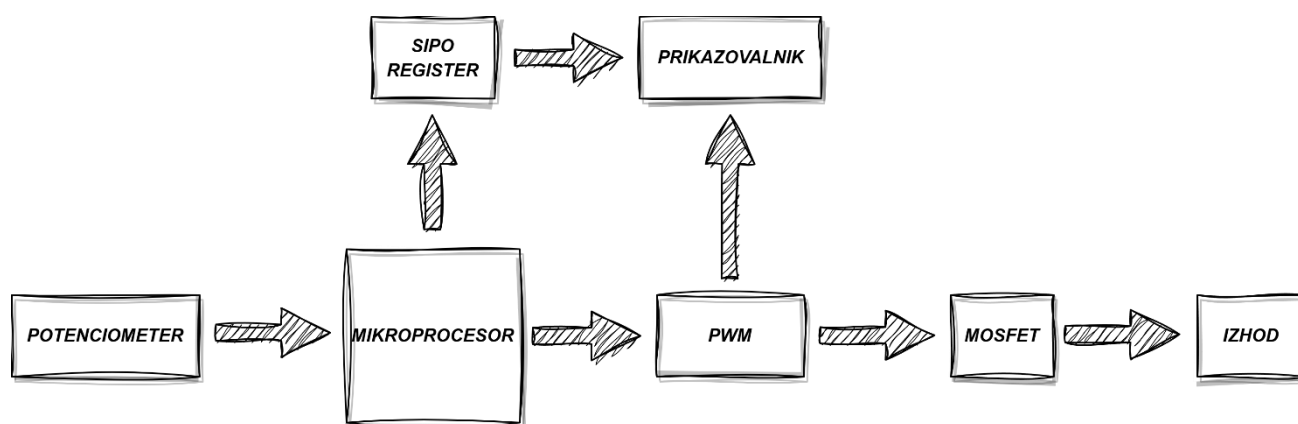
Ko so bile glavne komponente izbrane je sledila izbira stranskih manj kritičnih komponent. Za krmiljenje sem se nagibal k mikroprocesorju. Razlog je enostavna in fleksibilna implementacija kontrolne logike. Odločil sem se za skupino procesorjev Attiny, specifično Attiny85, saj mi je zelo blizu in izpolnjuje vse moje potrebe. Naletel sem le na manjšo težavo s številom izhodov, ki sem jo zelo preprosto odpravil z SIPO registerjem. Dodatno sem se odločil še za zatemnjevanje prikazovalnika.

Seveda to vezje potrebuje še napajanje. Zahteva je bila napajati vezje le z enim virom napetosti 12 V. Za pretvorbo vhodne napetosti v napetost, ki jo potrebuje krmilno vezje je bilo potrebna izbira med linearnim in digitalnim regulatorjem. Zaradi višje učinkovitosti sem se nagibal k digitalnemu step-down pretvorniku. Njegov način delovanja uporablja PWM signal, tuljavo in povratno vezavo. Prilagaja širino pulzov glede na to, kakšna je trenutna izhodna napetost, dokler ne doseže željene vrednosti. Napetost je nato zglajena z kondenzatorji. Več podrobnosti o takšnem regulatorju je navedenih na podatkovnem listu čipa.

Vse izbrane komponente je bilo potrebno poiskati še v spletni trgovini Farnell, preveriti razpoložljivost ter sešteti ceno, saj je omejitev nakupa znašala 20 €. Obenem pa še sem skopiral kode izdelkov. Te sem uporabil v seznamu elementov (BOM), preko katerega izvemo katere komponente, koliko kosov in kakšne vrednosti elementov je potrebno naročiti za izdelavo tiskanega vezja.

2.2 Blokovni diagram

Idejo je bilo potrebno še zapisati v obliki blokovnega diagrama. Iz njega lahko razberemo grobo delovanje vezja brez specifičnih komponent ali povezav. Je minimalna shema s katero si lahko veliko pomagamo pri risanju vezja v CAD programu in pri razumevanju delovanja. V veliko pomoč je tudi kadar elektronika ne deluje kot smo si zamislili in jo je potrebno popraviti. V diagramu so zajeti vhodi, glavne stopnje pretvorbi signalov, krmilna vezja ter izhodi. Ne smemo pozabiti na smeri poteka signalov.



Slika 1 Blokovni diagram vezja

Iz izrisanega diagrama izvemo, da je vezje krmiljeno s pomočjo potenciometra. Ta je priključen na mikroprocesor, ki je glavni element za obdelavo podatkov in krmiljenje sistema. Iz procesorja se signali porazdelijo. Na enem koncu imamo opravka z PWM signali. To so izhodi procesorja, na katerih je možno generirati pulzno širinsko modulacijo. V tem primeru imamo dva PWM izhoda. En je priključen na močnostni tranzistor (MOS-FET). Ta krmili izhod, saj iz procesorja samega ne moremo dobiti zadostnega toka ter napetosti za krmiljenje LED sijalk. Drug priključek z možnostjo pulzno širinske modulacije pa je uporabljen za zatemnjevanje celotnega prikazovalnika. To omogoča prilagajanje osvetljenosti prikazovalnika, da lahko zmanjšamo porabo in naredimo celotno napravo manj izstopajočo iz okolice. Prikazovalnik pa samo z PWM signalom še ne deluje. Tukaj nastopi procesor, ki ukazuje kaj se na prikazovalniku dogaja. Ker uporabljen procesor nima zadostnega števila priključkov, je za povezavo uporabljen vmesnik SIPO. SIPO je čip, ki nam pretvori serijski signal v 8 bit paralelnega.

3 Altium designer

3.1 Shema

Po izdelavi blokovnega diagrama in izbiri elementov sledi načrtovanje sheme vezja. Za ta korak sem uporabil Altium designer. Prvi korak je bil izdelati ali najti knjižnice za komponente. Če so komponente standardne in se uporabljajo v več projektih je smiselno knjižnico za to komponento izdelati sam ali jo pridobiti s spleta ter jo dodelati tako, da je kar seda dobro narejena za uporabo še v naslednjih projektih. Vse ostale so po lastni izbiri. V moji nalogi sem v večini uporabil že izdelane ter jih priredil mojim potrebam. Ta način mi je zelo skrajšal čas za izdelavo elektronike. Prednost knjižnic s spleta je tudi dodan 3D model komponente, ki ga je samemu težko in zamudno izdelati.

Ko sem uvozil vse knjižnice je sledila postavitvev in vezava elementov med seboj. Pri povezovanju in izbiri pravih komponent na napajalnem delu je prišla zelo prav standardna shema, pridobljena v podatkovnem listu čipa. Iz nje lahko vidimo katere komponente potrebuje čip za delovanje in kako so med seboj povezane. Za vrednosti elementov je bilo potrebnih tudi nekaj izračunov, ki so prav tako objavljeni v podatkovnem listu. Glavni vrednosti pridobljeni z izračunom sta bili upornosti uporov za povratno vezavo. Podatek o tem kakšna induktivnost tuljave je potrebna je bilo mogoče razbrati kar iz tabele.

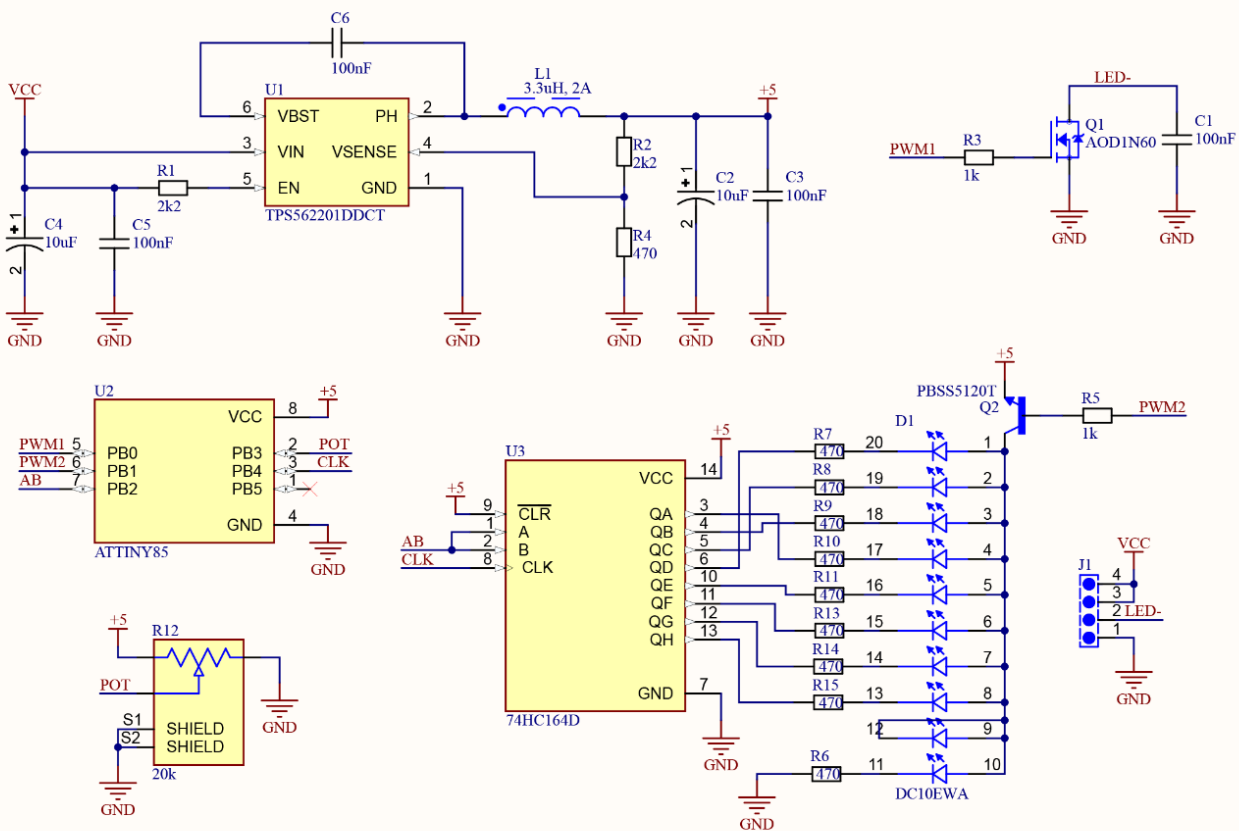
$$V_{out} = 0,768 * (1 + \frac{R1}{R2})$$

Izračun izhodne napetosti regulatorja

Sledili so izračuni za ostale komponente. Za uporabo LED prikazovalnika je potrebno zaporedno vezati upore za omejevanje toka. Njihove vrednosti je mogoče izračunati na podlagi formule, pri čemer I_d predstavlja maksimalen tok skozi diodo, U_{vcc} napajanje, U_d padec napetosti na diodi ter R predstavlja upornost predupora. Pri izbiri predupora sem izbral primeren tok skozi posamezno diodo katerega vrednost je bila 5 mA. Za to nalogo sem uporabil upore vrednosti 470 Ω saj je njihova standardna vrednost še najbližje izračunani. Pri izbiri preduporov za tranzistorja, velika natančnost ni bila potrebna. Zagotoviti je bilo samo potrebno polno odprtje tranzistorjev, saj v tem primeru delujeta kot stikalo. Na izhodu MOS-FET tranzistorja sem dodal še kondenzator za glajenje napetostnih špic.

$$R = \frac{U_{vcc} - U_d}{I_d}$$

Izračun predupora svetlečih diod



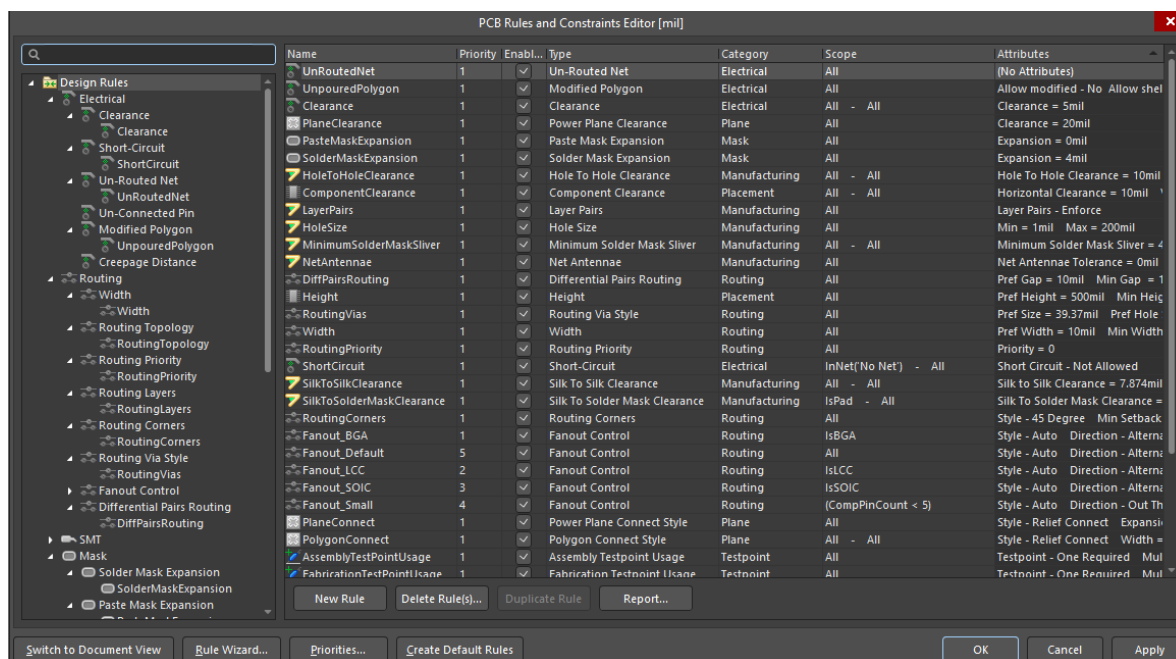
Tim Kosec

Slika 2 Shema vezja z izbranimi vrednostmi elementov

3.2 Tiskanina

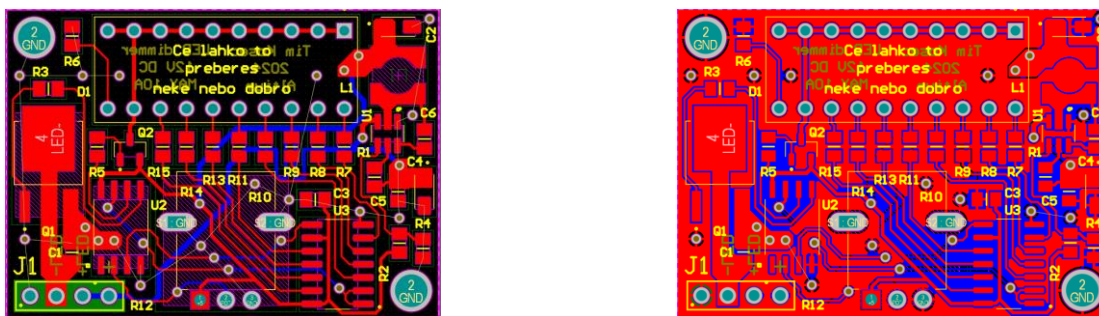
Po končani shemi se je lahko pričelo načrtovanje tiskanine. Pred izdelavo tiskanega vezja, je bilo potrebno uvoziti oziroma narisati podnožja elementov ter jih povezati z elementi v shemi. Za izdelavo teh je namenjen zavihek PCB_lib. V njem je mogoče izdelati in urejati podnožja elementov. Ko je podnožje izrisano, ga lahko povežemo z shemo in mu določimo parametre ter 3D model. Ko je bila celotna knjižnica pripravljena sem se lotil postavitve elementov na prazno ploščico. Še preden sem začel pa sem moral nastaviti še omejitve proizvajalca.

Omejitve proizvajalca so nastavitve, kjer določimo kakšne sposobnosti ima naš dobavitelj tiskanih vezij. Za pridobitev podatkov je potrebno odpreti njegovo spletno stran ter najti zmogljivosti. V altiumu odpremo zavihek design -> rules ter vstavimo vse parametre podane s strani dobavitelja. S tem smo zagotovili, da naše tiskano vezje ne bo kršilo nobenih pravil proizvajalca in ga bo možno izdelati.

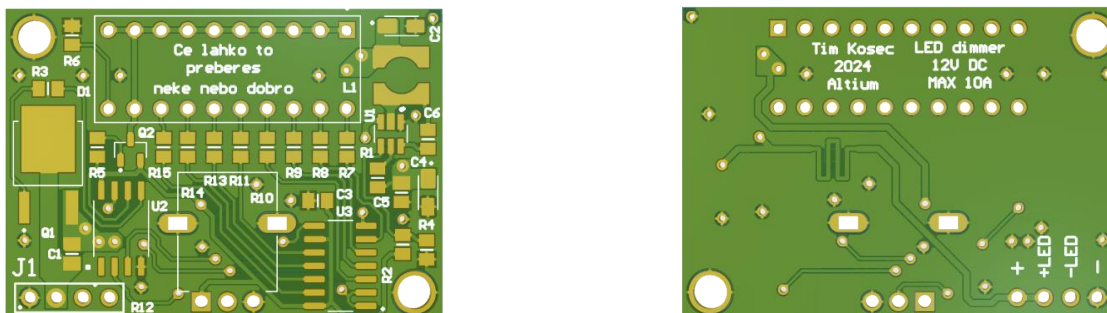


Slika 3 Zavihek s pravili dizajniranja tiskanine

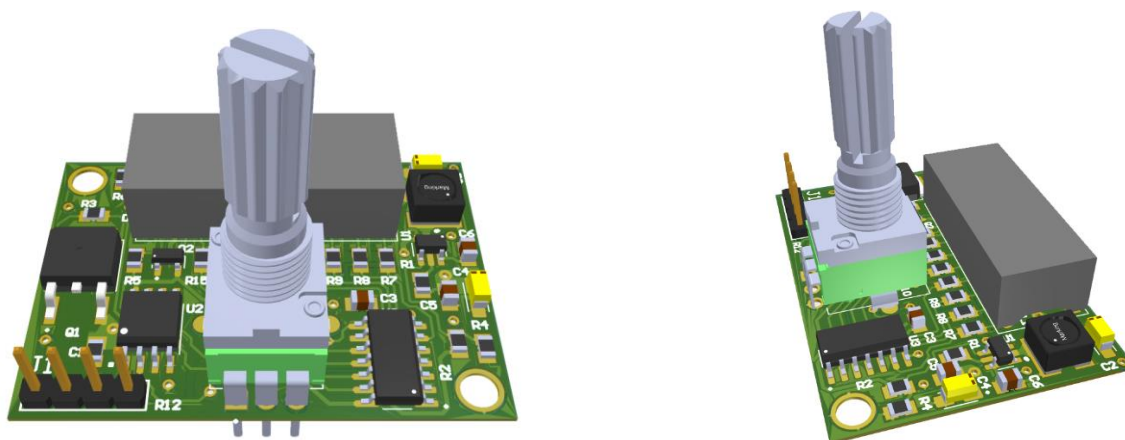
Sledil je uvoz elementov v PCB urejevalnik. Postavitev je bilo potrebno smiselno razporediti ter izbrati pravo velikost tiskanega vezja. Izbrati je bilo potrebno še število plasti. V tem primeru sta 2 plasti bili dovolj saj vezje ni preveč kompleksno in ni potrebe po ščitenju signalov pred motnjami. Sledilo je povezovanje komponent med seboj. Debelina povezav se razlikuje zaradi različnih zahtev po velikosti tokov. Signalne povezave so debeline 10 mils. Za napajalne povezave komponent je pa bilo smiselno uporabiti debelino 15 mils, saj s tem zmanjšamo padec napetosti na sami povezavi. Regulator napetosti je imel še dodatne zahteve po postavitvi komponent in povezavah, katere so opisane v podatkovnem listu. V projektu so tudi uporabljene vije, ki povezave obeh plasti povežejo med seboj. Ko so bile vse komponente povezane med seboj, se je lahko pričelo polivanje (polygon pour). Ta korak ni obvezen je pa priporočljiv. Z polivanjem v večini primerov združimo vse mase na vezju ter s tem omogočimo lepo porazdeljeno referenčno točko in odpravimo presluševanje med povezavami (crosstalk). Če je uporabljenih več kot dve plasti, lahko maso na ta način združimo v notranjih plasteh. Sledilo še je samo označevanje komponent. Na »top overlay« se nahaja plast, kjer so označene komponente in je tudi uporabljena za tiskanje besedil na tiskanino. Uporabljeni so tudi za označevanje kje se kakšna komponenta nahaja, kako je obrnjena ter kakšno oznako se imenuje. V veliko pomoč je tudi pri odpravljanju napak, saj po imenu takoj najdemo komponento tudi v shemi.



Slika 4 Tiskano vezje v urejevalniku



Slika 5 2D pogled tiskanega vezja



Slika 6 3D pogled tiskanega vezja

3.3 Seznam komponent

Izdelati je bilo potrebno še seznam komponent. Na njem so navedene vse uporabljene komponente v vezju. Iz njega se da razbrati oznaka elementa, njeno številko označeno na tiskanem vezju, podnožje, količino, dobavitelja in številko elementa, pridobljeno pri dobavitelju. Za generiranje seznama je potrebno vsaki komponenti določiti potrebne parametre.

Comment	Designator	Footprint	Quantity	Manufacturer	PartNumber
100nF	C1, C3, C5, C6	CAPC2012X140N	4	FARNELL	3013476
10uF	C2, C4	CAPPM3216X180N	2	FARNELL	2333010
DC10EWA	D1	DIP750W50P254L2540H825Q20	1	FARNELL	2290323
6130XX11121	J1	61300411121	1	FARNELL	3049569
3.3uH, 2A	L1	WE-HEPC_5030	1	FARNELL	2288159
AOD1N60	Q1	TO228P1003X240-3N	1	FARNELL	2454077
PBSS5120T	Q2	SOT95P230X110-3N	1	FARNELL	8736391
2k2	R1, R2	CAPC2012X140N	2	FARNELL	2447623
1k	R3, R5	CAPC2012X140N	2	FARNELL	1652937
470	R4, R6, R7, R8, R9, R10, R11, R13, R14, R15	CAPC2012X140N	10	FARNELL	9237445
20k	R12	TRIM_P0915N-QC20BR5K	1	FARNELL	1191742
TPS562201DDCT	U1	SOT95P280X110-6N	1	FARNELL	3121817
ATTINY85	U2	SOIC127P798X216-8N	1	FARNELL	1455164
74HC164D	U3	SOIC127P600X175-14N	1	FARNELL	1085337

Slika 7 Seznam komponent (BOM)

3.4 Gerber

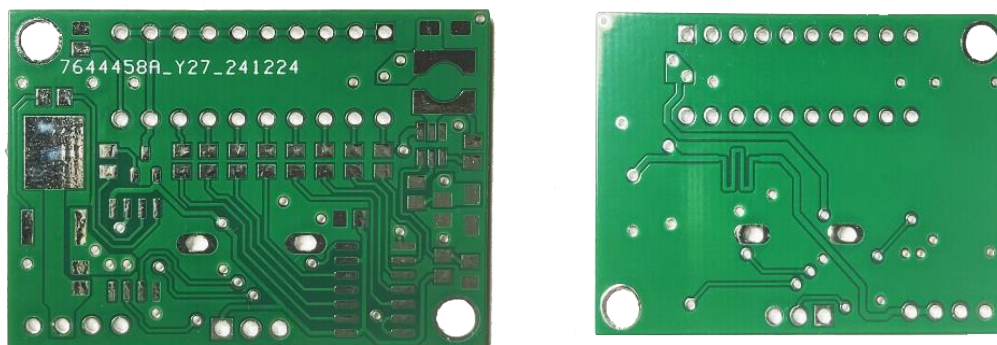
Ko je bilo delo v programu zaključeno, še je ostal le izvoz datoteke. Vrsta datoteke je gerber. Vsebuje vse informacije o tiskanem vezju in proizvajalcu omogoča izdelavo vezja po naročilu. V datoteki je opisana velikost vezja ter oblika. Tam najdemo tudi podatke o vijah ter luknjah, ki jih je v proizvodnji potrebno izvrtati. Seveda je tudi podana shema vseh povezav za vsako plast posebej ter dekorativni tisk. V nekaterih primerih je dodan tudi seznam kršitev pravil s katerim preverimo ali je tiskano vezje pravilno zasnovano.

4 Sestava elektronike

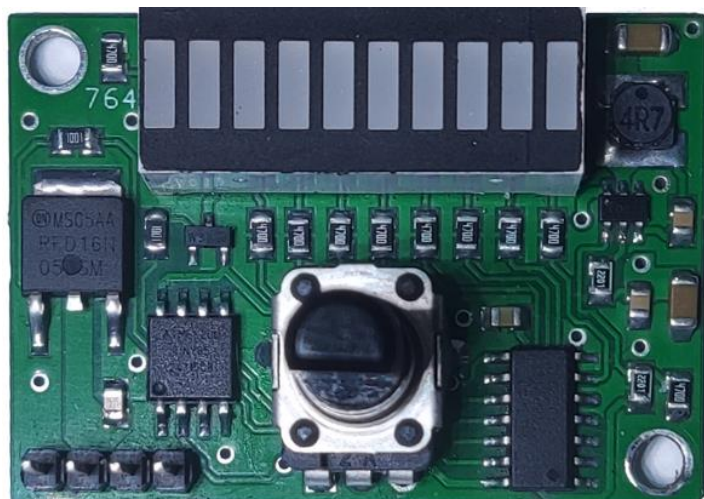
4.1 Sestava

Po naročilu tiskanih vezij in komponent je sledil del, kjer je bilo potrebno elektroniko sestaviti. Naročena vezja so prišla relativno hitro. Po pregledu kakovosti ter potrditvi, da se vse ujema s shemo sem ugotovil da imam eno manjšo težavo. Na tiskanem vezju ni bilo oznak komponent. To ni vplivalo na delovanje, saj so manjkale oznake, ki sem jih lahko poiskal v risbi tiskanine.

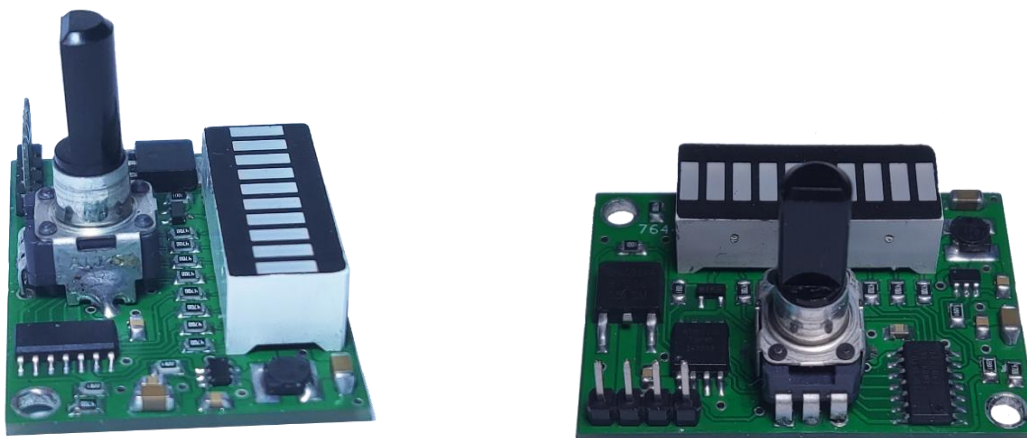
Sledila je postavitev elementov. S pomočjo tiskanega vezja v programu ter sheme sem lahko vse elemente brez težav spajkal na tiskanino. Težav ni delala nobena komponenta. Pred spajkanjem procesorja je bilo potrebno naložiti še program (več o tem v nadaljevanju). Ko so bile vse komponente pritrjene še je sledil pregled spojev in čiščenje paste z elektronike. S tem je tiskano vezje pripravljeno na naslednji korak, ki je testiranje ter posledično za uporabo v namen zakaj je bilo ustvarjeno.



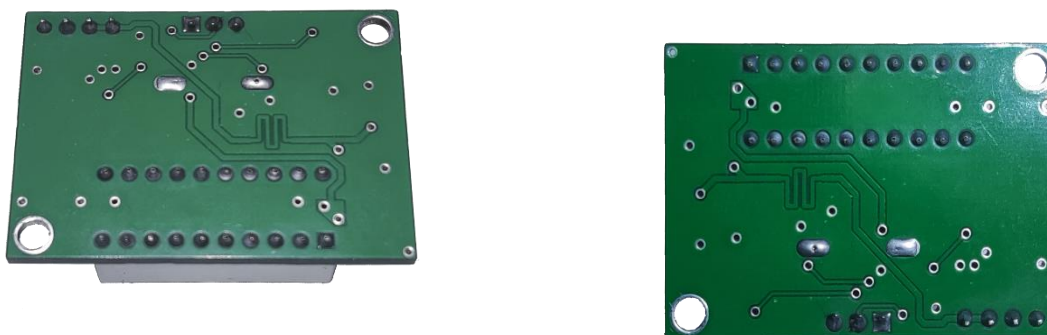
Slika 8 Tiskano vezje brez elementov



Slika 9 Končano tiskano vezje



Slika 10 Stranski pogled končanega tiskanega vezja

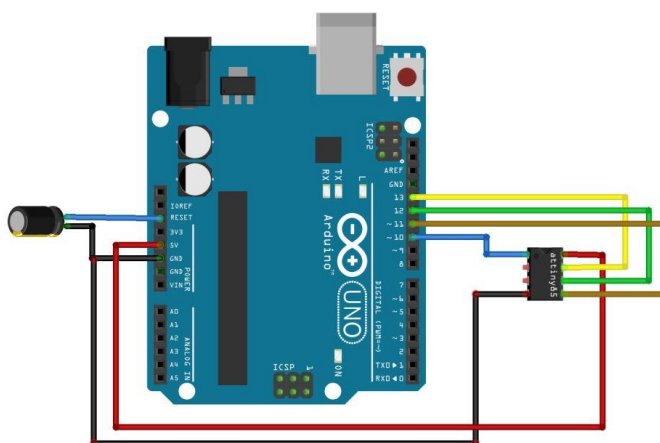


Slika 11 Pogled končanega tiskanega vezja od spodaj

4.2 Programiranje

Za delovanje je bil potreben še ključen element. Programiranje mikroprocesorja. V vezju sem uporabil procesor proizvajalca Atmel, družine Atmega, specifično Atmega 85. Vsebuje vse potrebno za moj projekt. PWM časovniki in ADC pretvornik sta v tem primeru zelo pomembna sestavna bloka tega čipa. Ti lastnosti sta pripomogli k izbiri tega čipa kot tudi nekaj drugih. Te so še, velikost spomina, kompaktnost ohišja, ravno pravšnje število priključkov ter poznavanje njegovega delovanja.

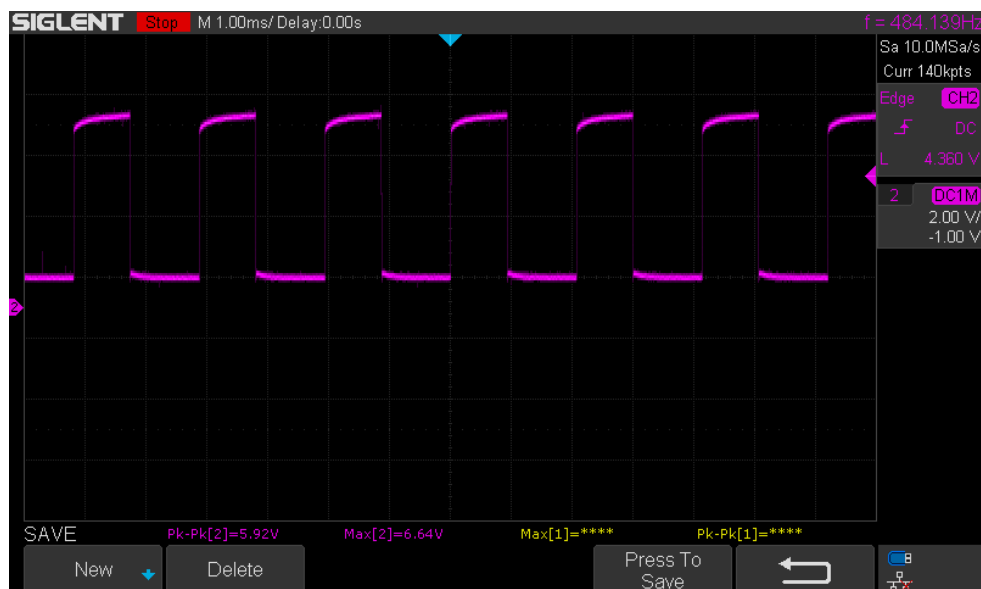
Predhodno sem napisal še program za delovanje celotne elektronike, ki sem ga tudi testiral z Arduino uno ter prototipno ploščico. Ko sem bil z programom zadovoljen je sledilo programiranje čipa na tiskanem vezju. Ta postopek je bil izveden s pomočjo Arduino uno razvojne ploščice, na katero je bilo potrebno predhodno naložiti program. Ta program omogoča prevajanje kode na računalniku v ISP protokol, s katerim je izbrani čip v vezju možno programirati. Po končanem nalaganju je bilo vezje končano in pripravljeno za testiranje.



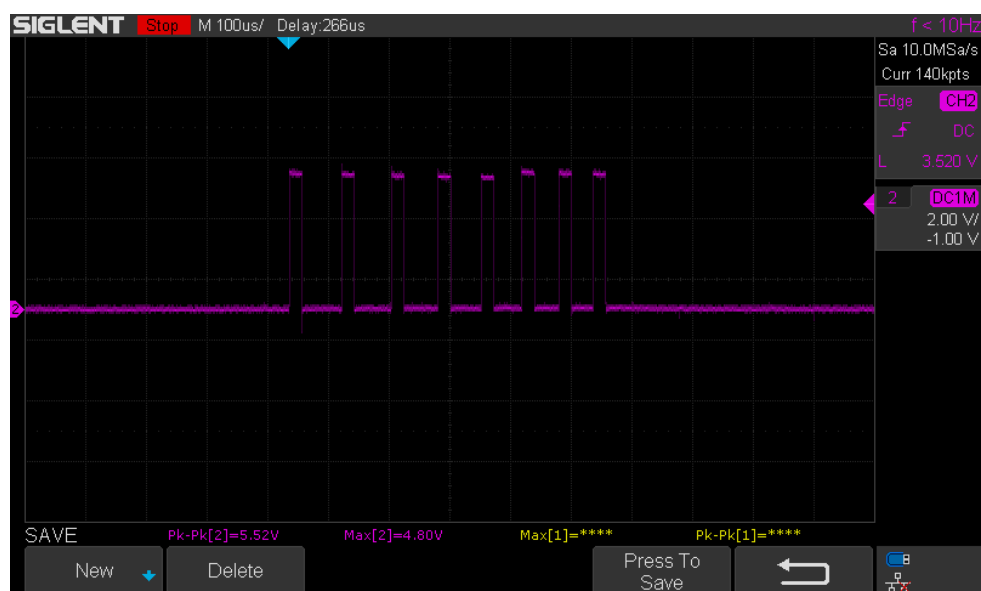
Slika 12 Priključitev procesorja za nalaganje programa

4.3 Testiranje

Sledilo še je samo testiranje elektronike. To je potekalo tako, da sem na njo priključil vse kar je bilo potrebno priključiti. Zraven sem vezal še osciloskop, s pomočjo katerega sem lahko opazoval signale na vezju ter videl ali se kje pojavljajo napetostne špice ter kako se vezje obnaša. V elektroniki nisem zasledil nobene težave ali pomanjkljivosti. Vezje je delovalo v prvem poskusu točno tako kot je bilo zasnovano.



Slika 13 Signal na izhodu vezja



Slika 14 Podatkovni signal SIPO registra

5 Opis delovanja

Opisana elektronika ima eno ključno nalogo. Zatemnjevanje enosmernih LED luči. To zagotavlja s spremembo izhodne napetosti. Za nastavitve svetlosti je uporabljen potenciometer. Ta pošilja analogni signal v mikroprocesor kjer z ADC pretvornikom pretvori analogni signal v digitalno obliko. Pridobljen signal obdela ter pošlje signal na izhod kot tudi na prikazovalnik. Prikazovalnik je priključen tako, da se na njem vklaplajo črtice v odvisnosti nastavitve svetlosti. Prva črtica ponazarja vklop vezja ter je vedno prižgana. Svetilnost je nastavljiva v 8 stopnjah. Dodatno je izvedeno tudi zatemnjevanje prikazovalnika. Izhod procesor regulira z pulzno širinsko modulacijo. To pomeni, da pri neki določeni frekvenci spreminja razmerje časa vključenega in izključenega signala. Na tak način lahko elektronika učinkovito ter enostavno regulira izhod ter posledično svetilnost luči. Samo vezje za napajanje uporablja step-down pretvornik.

6 Zaključek

Izdelava elektronike je potekala dokaj gladko. Težave s katerimi sem se srečal so bile večino usmerjene v Altium designer, saj je to moj prvi projekt izdelan v tem programskem okolju. Z samo elektroniko ni bilo zapletov. Tiskano vezje deluje točno tako kot v ideji. Tudi motnje okolice ne vplivajo na delovanje ter samo vezje ne oddaja nobenih problematičnih motenj v okolico.

Naslednji korak tega projekta bi sledil izdelava ohišja. Potrebno bi bilo izrisati 3D model ohišja v CAD programu ter ga naročiti pri proizvajalcu ali ga natisniti z 3D tiskalnikom. Če bi želel ta izdelek poslati na trg, bi potreboval zanesljivega dobavitelja komponent ter tiskanih vezij kot tudi dobavitelja za ohišja. Za resnejšo proizvodnjo bi morale vezje prestat še raznorazna testiranja ter pridobiti glavne certifikate. Tudi navodila so pri takšni stvari zelo pomembna.

7 Program

```

const int CLK = 4;
const int SDT = 2;
const int POT = A3; //3
const int OUT = 0;
const int DIM = 1;

int poten = 0;
int old_poten = 0;
int timer = 0;
int OUTval = 0;
int OUTval2 = 0;
int i = 0;
int k = 0;
int r = 0;

int PWM[9] = {0, 8, 16, 32, 64, 90, 140, 200, 255}; //PWM
vrednost posamezne stopnje

int dispVal = 0;

void setup() {
  pinMode(CLK, OUTPUT);
  pinMode(SDT, OUTPUT);
  pinMode(OUT, OUTPUT);
  pinMode(DIM, OUTPUT);
  pinMode(POT, INPUT);

  digitalWrite(CLK, LOW);
  digitalWrite(SDT, LOW);
  digitalWrite(OUT, LOW);
  digitalWrite(DIM, LOW);

  //Serial.begin(115200);

  dispVal = 0;
  tabela();
  digitalWrite(DIM, HIGH);
}

void tabela(){
  switch(dispVal){

case 0:           //87651234
  digitalWrite(DIM, HIGH);
  shiftOut(SDT, CLK, MSBFIRST, B11111111);
  digitalWrite(DIM, LOW);
  break;

case 1:
  digitalWrite(DIM, HIGH);
  shiftOut(SDT, CLK, MSBFIRST, B01111111);
  digitalWrite(DIM, LOW);
  break;

case 2:
  digitalWrite(DIM, HIGH);
  shiftOut(SDT, CLK, MSBFIRST, B00111111);
  digitalWrite(DIM, LOW);
  break;

case 3:
  digitalWrite(DIM, HIGH);
  shiftOut(SDT, CLK, MSBFIRST, B00011111);
  digitalWrite(DIM, LOW);
  break;

case 4:
  digitalWrite(DIM, HIGH);
  shiftOut(SDT, CLK, MSBFIRST, B00001111);
  digitalWrite(DIM, LOW);
  break;

case 5:
  digitalWrite(DIM, HIGH);
  shiftOut(SDT, CLK, MSBFIRST, B00001110);
  digitalWrite(DIM, LOW);
  break;

case 6:
  digitalWrite(DIM, HIGH);
  shiftOut(SDT, CLK, MSBFIRST, B00001100);
  digitalWrite(DIM, LOW);
  break;

case 7:
  digitalWrite(DIM, HIGH);
  shiftOut(SDT, CLK, MSBFIRST, B00001000);
  digitalWrite(DIM, LOW);
  break;

case 8:
  digitalWrite(DIM, HIGH);
  shiftOut(SDT, CLK, MSBFIRST, B00000000);
  digitalWrite(DIM, LOW);
  break;
  }}

```



```

uint16_t pridobiUrejenoVrednost(uint16_t vhodnaVrednost)
{
    const uint16_t odstopanje = 20;
    const uint16_t stevilolzgodnihStanj = 8;
    const uint16_t stanja[stevilolzgodnihStanj + 1] = {0, 130,
222, 348, 474, 600, 726, 842, 948};
    const uint16_t zacetnoStanje = 0;
    static uint16_t trenutnoStanje = zacetnoStanje;
    uint16_t k = stanja[trenutnoStanje];

    if (trenutnoStanje > 0) {
        k -= odstopanje;
    }
    uint16_t j = stanja[trenutnoStanje + 1];

    if (trenutnoStanje < stevilolzgodnihStanj){
        j += odstopanje;
    }

    if (vhodnaVrednost < k || vhodnaVrednost > j) {
        uint16_t i;

        for (i = 0 ; i < stevilolzgodnihStanj ; i++) {
            if (vhodnaVrednost >= stanja[i] && vhodnaVrednost <=
stanja[i + 1]) {
                break;
            }
        }

        trenutnoStanje = i;
    }

    return trenutnoStanje;
}

void loop() {

    poten = pridobiUrejenoVrednost(map(analogRead(POT), 0,
1024, 1024, 0));

    if(poten != old_poten){
        old_poten = poten;
        dispVal = poten;
        tabela();
        timer = 400;
        digitalWrite(DIM, LOW);
    }

    timer--;
    if(timer<=1){
        timer=1;
        analogWrite(DIM, 230);
    }

    OUTval = PWM[poten];

    if(OUTval != OUTval2){
        if(OUTval < OUTval2){
            r = 1;
            k = OUTval;
        }
        if(OUTval > OUTval2){
            r = 2;
            k = OUTval;
        }
        if(OUTval == OUTval2){
            r = 0;
            i = OUTval;
        }
        OUTval2 = OUTval;
    }

    if(r == 1){
        if(i > k){
            i--;
            analogWrite(OUT, i);
        }
    }

    if(r == 2){
        if(i < k){
            i++;
            analogWrite(OUT, i);
        }
    }

    delayMicroseconds(1000);
}

```