

Kazalo

1	Povzetek	4
2	Uvod.....	5
3	Hipoteza.....	6
4	Vsebinski del.....	7
3.1	Opis komponent	7
3.1.1	Fotoupor.....	7
3.1.2	Krmilnik Alpha Mitsubishi	8
3.2	Izdelovanje programa za krmiljenje ulične razsvetljave in semafor	11
3.2.1	program za ulično razsvetljavo.....	11
3.2.2	Program za semafor.....	13
3.3	Odštevalnik časa	14
3.3.1	7-segmentni prikazovalnik.....	15
3.3.2	BCD/7-segmentni dekoder.....	16
3.3.3	Dekadni števec.....	17
3.3.4	Časovnik NE 555	18
3.4	Izdelovanje tiskanine	20
3.5	Načrt za izdelovanje makete	23
5	Družbena odgovornost.....	24
6	Sklep.....	25
7	Viri.....	26

Kazalo slik

Slika 1: Padec napetosti na fotouporu v odvisnosti od svetlobe.	7
Slika 2: Zatemnjen fotoupor.....	8
Slika 3: Osvetljen fotoupor.....	8
Slika 4: Krmilnik Alpha Mitsubishi [1].	8
Slika 5: Programiranje časovnega funkcijskega bloka Time Switch.	10
Slika 6: Program za krmiljenje ulične razsvetljave in semafor.	11
Slika 7: Ulična razsvetljava [2].	11
Slika 8: Semafor [3].	13
Slika 9: Prvi poizkus programa za krmiljenje semaforja.....	14
Slika 10: 7-segmenti prikazovalnik.	15
Slika 11: Prikaz števil od 9-0 na 7-segmentnem prikazovalniku.	16
Slika 12: Prikaz BCD, 7-segmentnega koda (K) in decimalna števila [4].	16
Slika 13: Dekoder in 7-segmentni prikazovalnik.	17
Slika 14: Dekadni števec IC 74192 v vezavi.	17
Slika 15: Astabilni multivibrator s časovnikom NE 555.	18
Slika 16: Napetost na kondenzatorju C1 in izhodni signal astabilnega multivibratorja.	19
Slika 17: Sestavljeno vezje na eksperimentalni ploščici.	20
Slika 18: Elektronski načrt generatorja impulzov in števca, sestavljen v okolju Eagle.	21
Slika 19: Pozicija elementov na ploščici tiskanega vezja.....	22
Slika 20: Načrt tiskanine.	22
Slika 21: Osnovne komponente makete.	23
Slika 22: Načrt makete.....	23

1 Povzetek

Raziskovalna naloga prikazuje, kako bi lahko varčevali z električno energijo pri ulični razsvetljavi in kako bi skrajšali nepotreben čas vklopljene razsvetljave, ko je zunaj že svetlo in je ulična razsvetljava še vedno vklopljena. Prva težava je bila, da sva morala zunaj zaznavati svetlobo. Druga težava pa je bila, da sva morala to nekako pretvoriti v podatek, ki ga bo lahko krmilnik razumel. Zadnja težava pa je bila ta, da sva morala ugotoviti, kako v tistem delu noči, ko zunaj ni veliko ljudi in avtomobilov, izklopiti polovico razsvetljave. Prvo težavo sva rešila s foto uporom. Drugo težavo sva rešila z uporabo analognega vhoda krmilnika. Napetost, ki jo neprestano čuti analogni vhod, krmilnik preoblikuje v digitalno obliko. Tretjo težavo pa sva enostavno rešila s pomočjo časovnika.

V izdelavi raziskovalne naloge sva tudi spoznala način krmiljenja semaforja, kako deluje in to upoštevala pri programiranju krmilnika. Dodala sva tudi števec, ki pešcem prikaže, koliko časa imajo na voljo za prečkanje cestišča. Pri programiranju sva morala paziti, da se stanje semaforja za pešce spremeni le takrat, če pešec pritisne na gumb, ki je nameščen ob prehodu. Ko nama je to uspelo, sva začela s števcem. Najprej sva poiskala ustrezna integrirana vezja, izdelala ter preizkusila delovanje vezja na eksperimentalni ploščici, nazadnje pa sva za vezje števca izdelala še tiskanino.

2 Uvod

Življenja v današnjem času si brez ulične razsvetljave ne moremo predstavljati, saj zelo olajša promet in vidljivost pešcev. Vendar pa večinoma ne gledamo na to, da bi z ulično razsvetljavo ravnali energetske varčno. Ta problem sva opazila, ko sva zjutraj hodila v šolo in je že bilo svetlo, ampak razsvetljava je še vedno bila vključena. Ulična razsvetljava je bila večkrat vključena po eno uro ali trideset minut, čeprav je že bilo dovolj svetlo. To naju je zelo motilo, saj so naju v šoli učili, da moramo z energijo vedno varčevati. Zato sva se odločila, da bova raziskovala, kako bi bolj učinkovito krmilila ulično razsvetljavo. Sploh v današnjem času, ko prehod na obnovljive vire še ni popoln, moramo varčevati s neobnovljivimi viri, saj nam jih zmanjkuje, pa tudi onesnažujejo naravo. Stroški za posodobitev se relativno hitro povrnejo in bi se lahko veliko privarčevalo, saj je dandanes velika denarna kriza. Za boljšo prihodnost je potrebno misliti na okolje, ki ga moramo čim manj obremenjevati.

3 Hipoteza

Opazila sva, da na nekaterih krajih uporabljajo za krmiljenje ulične razsvetljave samo uro, vendar je ta način problematičen zaradi letnih časov. Pozimi je dan daljši, poleti pa krajši, zato je potrebno nekakšno prilagajanje na ta cikel. Ugotovila sva, da bi lahko ta problem najlažje rešili z uporabo senzorja svetlobe, fotoupora LDR (Light dependent resistor), v povezavi z uro. Ponoči ob določenem času, ko je manj prometa in manj pešcev, se polovica ulične oziroma cestne razsvetljave izklopi. Pločniki in cesta bi bili še dovolj osvetljeni. Tako bi najbolj učinkovito varčevali z energijo.

Varčevanje:

- krmiljenje z uro
- krmiljenje s svetlobnim senzorjem

Krmilna oprema:

- krmilnik Alpha Mitsubishi

Cilji:

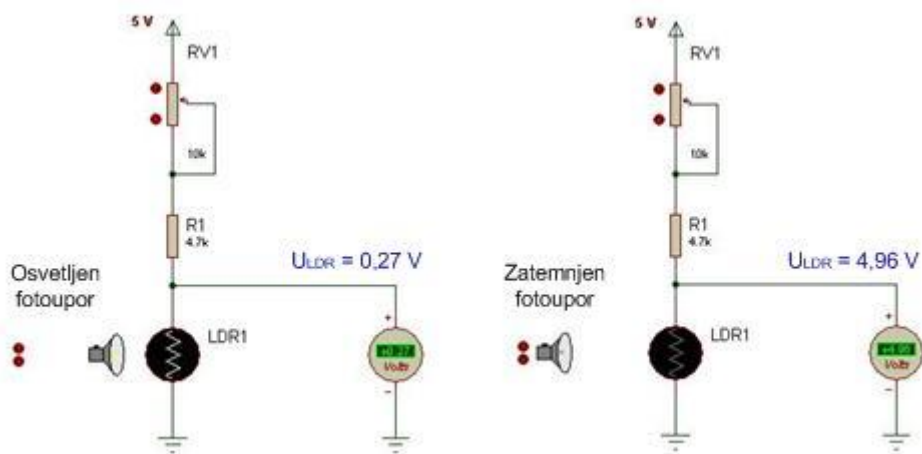
- lažje krmiljenje ulične razsvetljave
- energetsko varčevanje pri ulični razsvetljavi
- pridobitev izkušenj in znanja pri programiranju krmilnika Alpha Mitsubishi

4 Vsebinski del

3.1 Opis komponent

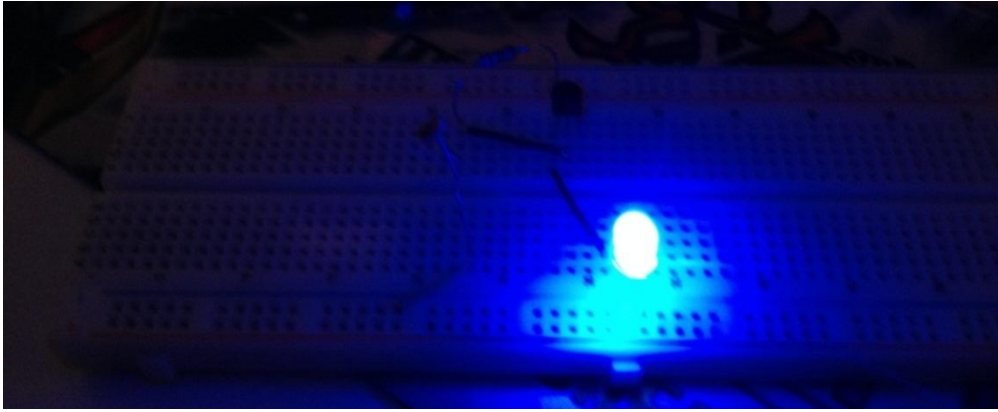
3.1.1 Fotoupor

Fotoupor LDR (ang. Light dependent resistor), je najpreprostejši svetlobni senzor, je pravzaprav polprevodniška ploščica v masivni ali tankoplastni izvedbi z ustrezno nameščenimi ohmskimi kontakti. Zgrajen je iz kadmijevega sulfida (CdS), njegova upornost je svetlobno odvisna. Ko svetloba vpade na njegovo polprevodniško površino, se v njem generirajo prosti elektroni in vrzeli, ki povečajo prevodnost foto upora. Tukaj je primer, kako povečevanje svetlobe spreminja napetost na uporovnem delilniku. Ker se upornost fotoupora z osvetlitvijo spremeni, se na njem spremeni tudi padec napetosti.

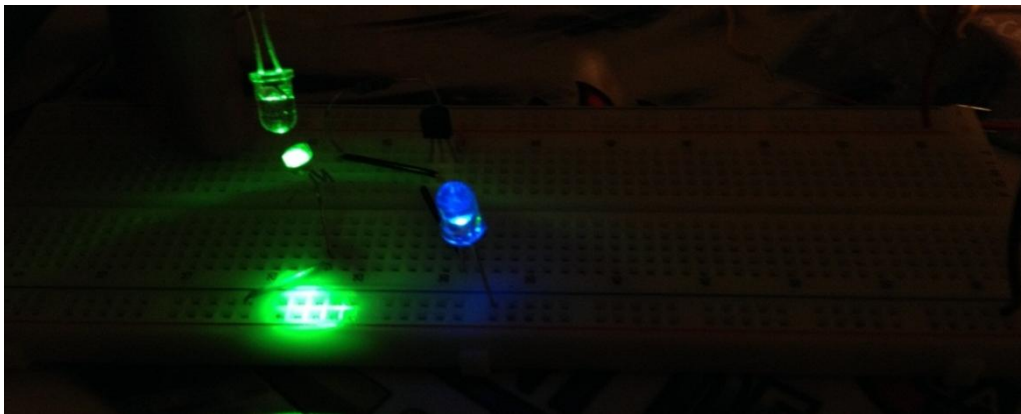


Slika 1: Padec napetosti na fotouporu v odvisnosti od svetlobe.

Da bi ugotovila delovanje fotoupora, sva naredila poskus in sicer krmiljenje LED diode v odvisnosti od svetlobe. Več svetlobe ki jo prejme fotoupor, slabše sveti LED dioda in obratno.



Slika 2: Zatemnjen fotoupor.



Slika 3: Osvetljen fotoupor

3.1.2 Krmilnik Alpha Mitsubishi



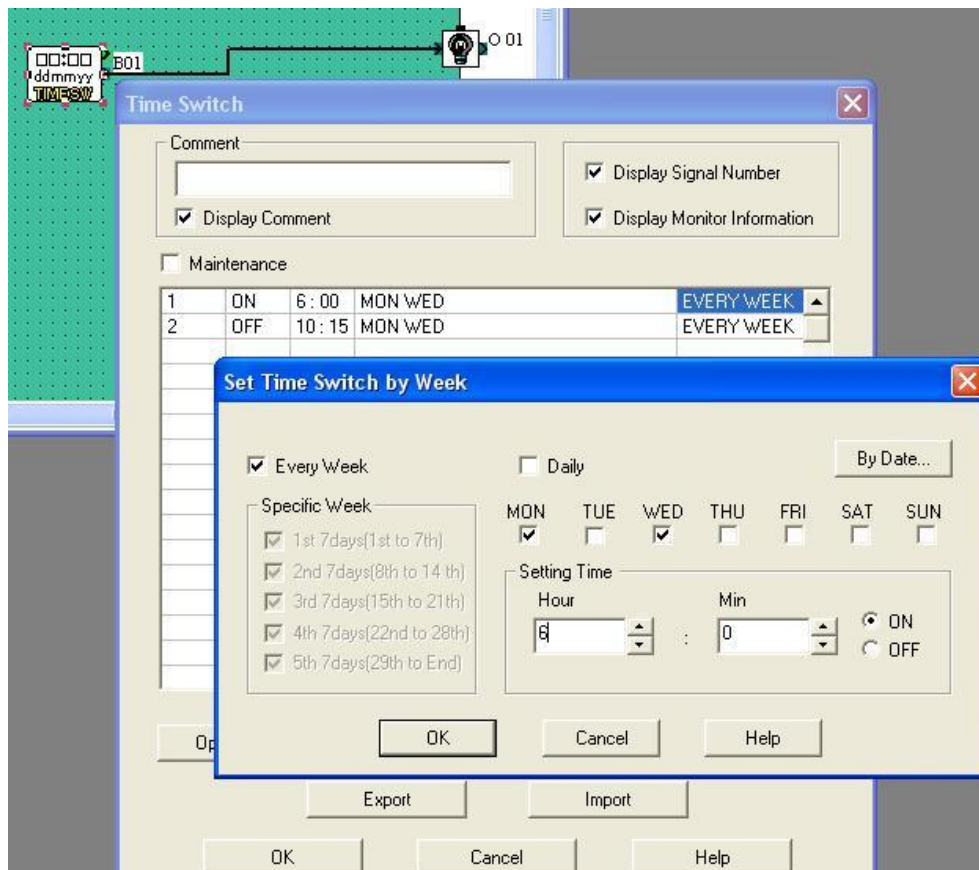
Slika 4: Krmilnik Alpha Mitsubishi [1].

Mitsubishi Alpha controller je krmilnik, ki nam omogoča krmiljenje raznih naprav. V najinem primeru ulične razsvetljave. Programira se na takšen način, da nam grafično prikaže vezje, ki ga mi sestavljamo. V delovno površino programskega okolja Alpha programming vstavljamo razna logična vrata in funkcijske bloke. Te ustrezno povezujemo, odvisno od zahtevanega krmilja. Ko izdelamo program, ga preizkusimo in s posebnim kablom prenesemo v krmilnik. Preden sva lahko začela izdelovati program, sva morala spoznati kar nekaj funkcijskih blokov. Začela sva s spoznavanjem osnovnih logičnih vrat. To so *AND*, *OR* in *NOT* logična vrata.

AND logična vrata dajo na izhod logično 1 le takrat, ko so vsi vhodi na logični 1. *NOT* logična vrata negirajo vhodno vrednost. To pomeni, da je na izhodu logična 1, če je vhod priključena logična 0 in obratno. *OR* logična vrata pa dajo na izhodu logično 1, če je vsaj eden od vhodov na logični 1.

Spoznala sva tudi zakasnitveni funkcijski blok *Delay*. Z njim lahko nastavljamo zakasnitve tako pri vklopu, kot pri izklopu. Pomembna funkcija, ki sva jo uporabljala, je bila *One Shot*. Prožimo jo s pritiskom na tipko. Dolžino signala na izhodu tega funkcijskega bloka določimo v priročnem meniju, do katerega pridemo z dvoklikom na funkcijski blok.

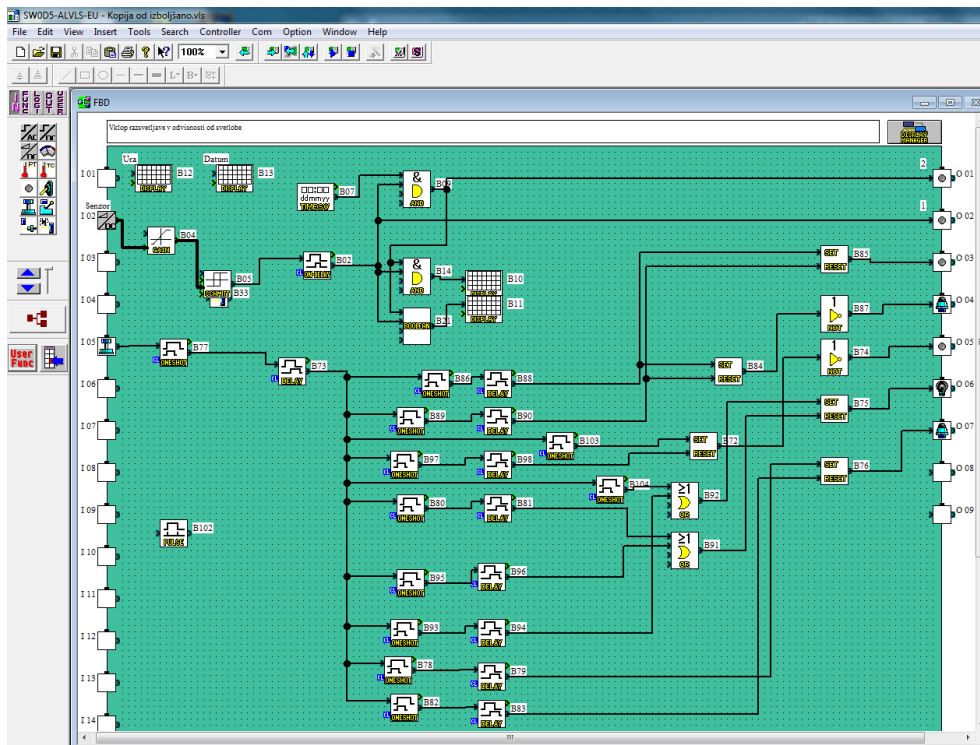
Zadnji pomemben element, ki sva ga morala spoznati je bil funkcijski blok *Time Switch*. Z njim sva določila čas po posameznih dnevih v tednu, ko se polovica razsvetljave izklopi.



Slika 5: Programiranje časovnega funkcijskega bloka Time Switch.

Na sliki 5 dobro vidimo, kako vnašamo podatke v priročnem meniju funkcijskega bloka *Time Switch*. Določimo lahko dan, uro in minute, ko naj se zgodi ustrezna sprememba. Na primer določimo lahko, da se bo v sredo ob 7 uri zvečer razsvetljava vklopila, nato pa v četrtek ob 5 uri zjutraj izklopila. Z uporabo zgoraj opisanih logičnih vrat ter funkcijskih blokov sva izdelala program krmilja za najino nalogo.

3.2 Izdelovanje programa za krmiljenje ulične razsvetljave in semafor



Slika 6: Program za krmiljenje ulične razsvetljave in semafor.

3.2.1 program za ulično razsvetljavo



Slika 7: Ulična razsvetljava [2].

Ulična razsvetljava mora biti odvisna od svetlobe, zato sva morala za ustrezno delovanje krmilja uporabiti senzor svetlobe. Nastala je težava, saj mora to, kar senzor

zazna, krmilnik razumeti in ustrezno uporabiti. Pri razrešitvi težave nama je pomagal mentor, saj je sama nisva znala rešiti.

Razložil nama je, da lahko podatke, ki jih zazna senzor, pripeljemo na analogni vhod krmilnika. Na analogni vhod lahko priključimo napetost od 0 VDC do 10 VDC. Analogno-digitalni pretvornik je 10 bitni, resolucija pa je pri 10 VDC 550. To pomeni, da analogni vhod zazna spremembo napetosti, ki je večja od 18,18 mV ($10 \text{ V}/550 = 18,18 \text{ mV}$). V programu povežemo analogni vhod s funkcijskim blokom *Gain*, tega pa z Schmittovim prožilnikom (*Schmitt Trigger*). Analogne povezave so širše, povezujejo pa se na zeleno obarvane vhodne oz. izhodne priključke funkcijskih blokov.

Program neprestano preverja, kakšna je napetost na analognem vhodu. Ko prekorači določeno (vpisano) vrednost, spremeni stanje izhodnega signala funkcijskega bloka *Schmitt Trigger*.

Sestavila sva vezje s fotouporom in določila tisto vrednost napetosti na fotouporu, pri kateri naj bi se vklopila razsvetljava. Po nekaj poizkusih sva ugotovila, da je ta vrednost 5,45 V, kar predstavlja bitni podatek 300, zato sva to vrednost vpisala v funkcijski blok *Schmitt Trigger*.

Prišla sva do naslednje težave, razsvetljava se ne sme takoj vklopiti oziroma izklopiti. Na senzor svetlobe namreč lahko za kratek čas posvetijo luči avtomobila, lahko pa se za kratek čas senzor svetlobe tudi zatemni (ptica in naredi senco). Krmilje zato ne sme takoj reagirati na takšne, hipne spremembe. Zato sva morala v krmilju uporabiti zakasnitve. Te sva nastavila na 10 sekund. Tako mora biti 10 sekund izpopolnjen pogoj, da se spremeni stanje v ulični razsvetljavi. V praksi je ta čas daljši in traja okoli ene minute.

Dobila sva idejo, da bi z uporabo časovnika *Time Switch* krmilje izdelala tako, da bi pri ulični razsvetljavi tudi varčevali z električno energijo. Med urami od polnoči do jutranjih ur, med tednom ni skoraj nihče zunaj, zato bi se lahko polovica ulične razsvetljave izklopila. Bila pa bi še vedno zadostna svetloba za promet. Morala sva se odločiti, kdaj se bo vsak dan polovica razsvetljave izklopila. Za ponedeljek, torek, sredo in četrtek sva izbrala uro 22:30, takrat zunaj večinoma ni nihče in avtomobilov je zelo malo. Za petek in soboto sva izbrala čas 23:55, saj po tej uri večina ljudi že spi, avtomobilov je najn in tudi če je kdo šel ven s prijatelji, so se večinoma že vrnili

domov. Za nedeljo pa sva izbrala čas 23:00, zaradi tega, ker so še vseeno nekateri malo dlje zunaj. Čase po dnevih, ki sva jih določila, veljajo čez vsa leto.

3.2.2 Program za semafor



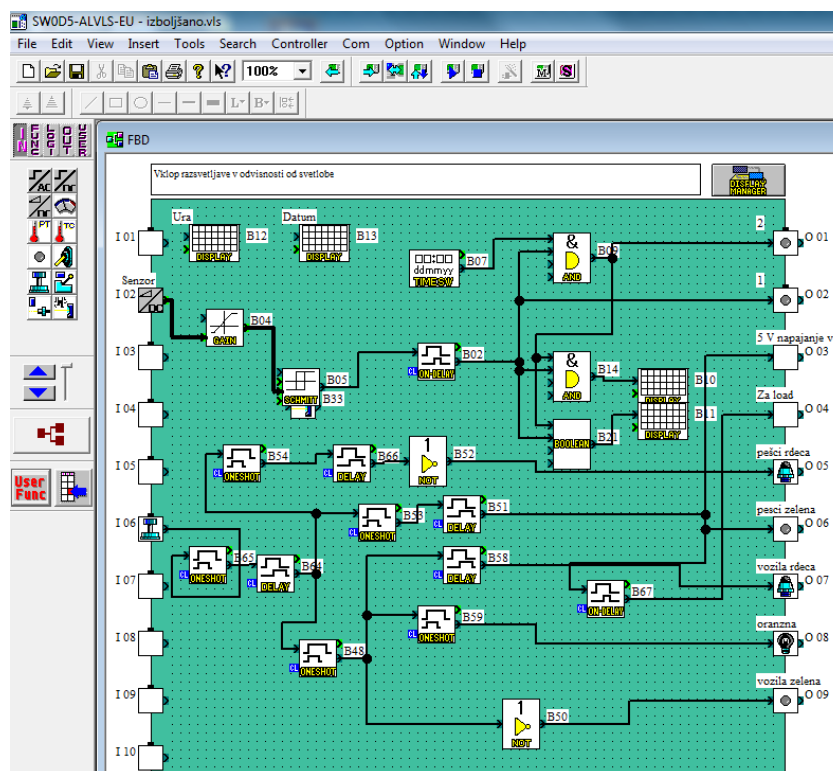
Slika 8: Semafor [3].

Začela sva razmišljati o izdelovanju makete in mentor nama je dal super idejo, da bi lahko maketa predstavljala enosmerno cesto s prehodom za pešce in s semaforjem. Začela sva z izdelovanjem programa za semafor. Najprej sva si zastavila pogoje oziroma zahteve, po katerih bo semafor deloval.

Ob semaforju za pešce sva namestila tipko, ki je namenjena sprožitvi cikla za prehod iz rdeče luči na semaforju v zeleno luč, namenjeno za prehod pešcev. Dokler pešec ne pritisne na tipko, poteka promet nemoteno. Voznikom sveti zelena, pešcem pa rdeča luč na semaforju. Ko pešec pritisne na tipko, se po 5-ih sekundah začne postopek spreminjanja stanja semaforja. Najprej se voznikom vklopi oranžna luč, čez določen čas pa rdeča luč. Ko vozniki stojijo pred rdečo lučjo, se sproži postopek za preklop semaforja za pešce iz rdeče v zeleno luč. Pri izdelavi krmilja sva morala uskladiti vse potrebne zakasnitve, da je semafor deloval pravilno. Ko pešci prečkajo cestišče, se mora stanje na semaforjih povrniti. Za pešce se nama je zdela stvar preprosta, za voznike pa sva morala upoštevati dejstvo, da nekaj časa svetita istočasno rdeča in oranžna luč, nakar se izklopita in se vklopi zelena luč na semaforju. To sva rešila z uporabo ustreznih logičnih vrat.

Potem so se pa začele težave, saj sva poskusila v krmilju uporabiti funkcijski blok *One Shot*, ki pa ima samo en vhod. To sva hotela popraviti s funkcijo *Set Reset*, a sva prišla na isto težavo, krmilje ni delovalo pravilno. Na koncu sva jo rešila z *OR* logičnimi vrati.

Nato se je pojavil problem, če je pešec večkrat pritisnil na tipko. Pojavile so se namreč motnje. Problem sva rešila tako, da sva dodala funkcijski blok *One Shot*, ki ostane vklopljen, dokler se krmilje ne izvrši do konca, saj med izvrševanjem programa, pešec ne mora vplivati na vhod funkcijskega bloka.



Slika 9: Prvi poizkus programa za krmiljenje semaforja.

3.3 Odštevalnik časa

Ko sva hodila iz šole, sva opazila na prehodu za pešce prikazovalnik, ki je prikazoval čas, ki ga ima pešec na razpolago, da prečka cesto. Takoj sva pomislila, da bi lahko to naredila tudi midva, a nisva vedela kako. Mentor nama je dajal, da bi lahko to naredila z uporabo dekadnega števca, dekoderja in 7-segmentnega prikazovalnika.

Za potrebne šteвне impulze pa bi lahko uporabila časovnik NE 555 v vezavi astabilnega multivibratorja. Sprva sva morala spoznati vse te elemente.

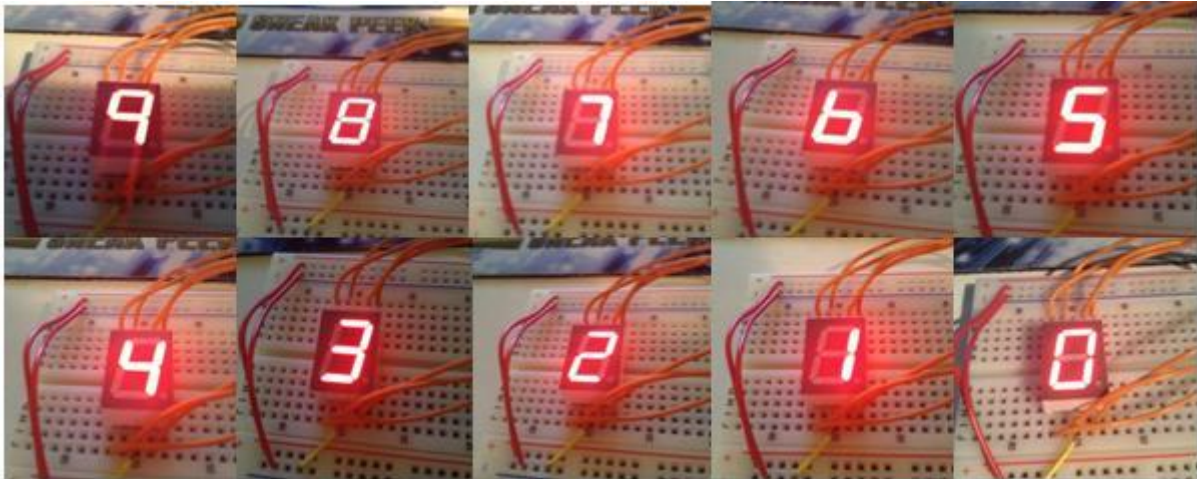
3.3.1 7-segmentni prikazovalnik



Slika 10: 7-segmentni prikazovalnik.

Poznamo dve osnovni vrsti 7-segmentnih prikazovalnikov. Prikazovalniki s skupno anodo in prikazovalniki s skupno katodo. Tisti s skupno anodo imajo anode vseh svetlečih diod (segmentov) povezane skupaj v en priključek, ki ga povežemo na pozitivno napetost. Tisti s skupno katodo pa imajo katode vseh svetlečih diod (segmentov) povezane skupaj v priključek, ki ga povežemo na maso.

Midva sva uporabila prikazovalnik s skupno anodo. 7-segmentni prikazovalnik vsebuje 10 priključkov. Deluje tako, da vsak segment predstavlja svetlečo diodo ki sveti, ko je vklopljen. Kateri segmenti svetijo je odvisno od številke, ki jo želimo prikazati. Segmenti so poimenovani, zgornji segment ima ime *a*, potem pa v smeri urnega kazalca naprej, seveda po angleški abecedi, do segmenta z imenom *f*, nazadnje še pa sredinski z imenom *g*. Če želimo prikazati število tri, morajo svetiti segmenti *a*, *b*, *c*, *d* in *g*. Slika 11 prikazuje primere, kako 7-segmentni prikazovalnik prikazuje števila od 9 do 0.



Slika 11: Prikaz števil od 9-0 na 7-segmentnem prikazovalniku.

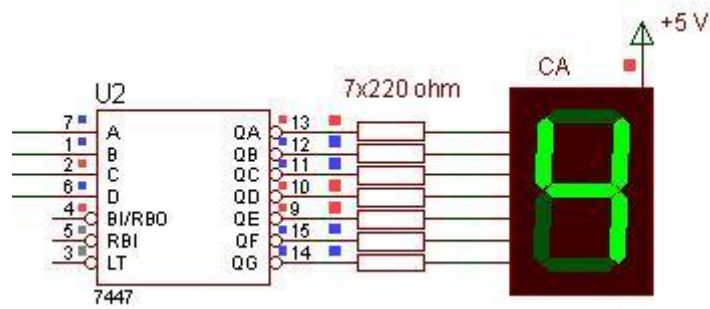
3.3.2 BCD/7-segmentni dekoder

Je v bistvu integrirano vezje, ki pretvori BCD kod v 7-segmentni kod.

BCD inputs				segment outputs							display
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	1
0	0	1	1	1	1	1	1	0	0	1	1
0	1	0	0	0	1	1	0	0	1	1	1
0	1	0	1	1	0	1	1	0	1	1	1
0	1	1	0	0	0	1	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0	1
1	0	0	0	1	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1	1

Slika 12: Prikaz BCD, 7-segmentnega koda (K) in decimalna števila [4].

Poznamo 7-segmentni kod za prikazovalnike s skupno katodo (K) in 7-segmentni kod za prikazovalnike s skupno anodo (A). Midva sva uporabila IC 7447, namenjen prikazovalnikom s skupno anodo. Ima štiri vhode (BCD kod) in sedem izhodov (a-g).

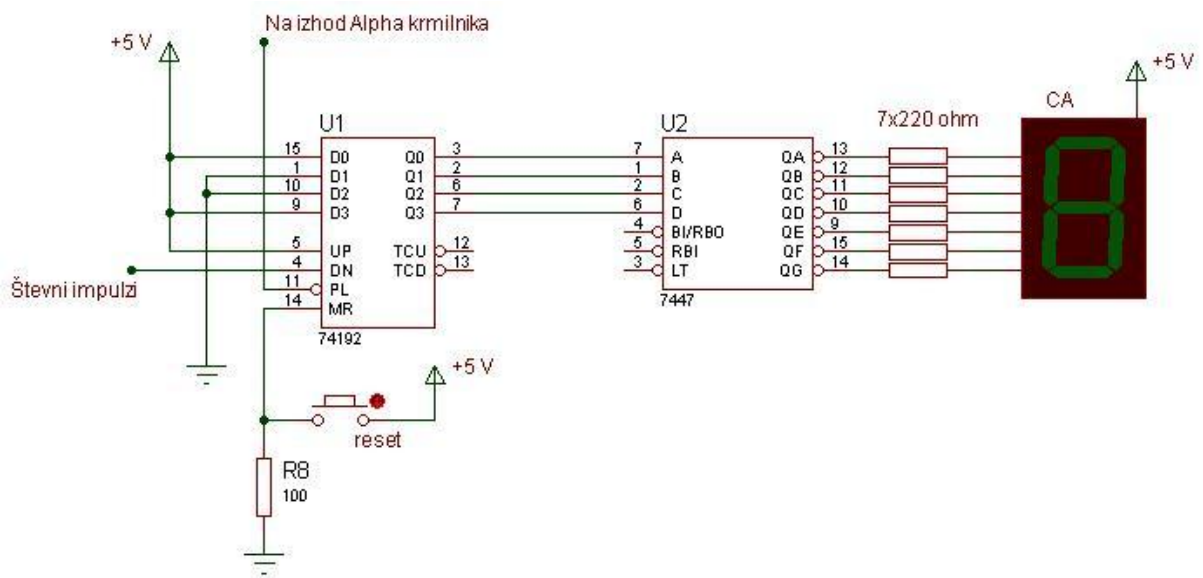


Slika 13: Dekoder in 7-segmentni prikazovalnik.

3.3.3 Dekadni števec

Dekadni števec IC 74192 ima dva števna vhoda, enega za štetje gor in drugega za štetje dol. Midva sva uporabila števni vhod za štetje dol, saj bo prikazovalnik pešcem prikazoval čas v sekundah od 9 do 0.

Vsebuje tudi podatkovne vhode D0, D1, D2 in D3, na katere pripravimo želeno število, ki ga števec prestavi na izhod, kadar je v stanju presetiranja. To se zgodi takrat, ko je priključek 11 (Load) prestavljen iz logične 1 v logično 0.



Slika 14: Dekadni števec IC 74192 v vezavi.

Midva krmiliva ta priključek z enim od izhodov krmilnika Alpha. Na začetku mora biti ta priključek v stanju logične 0, da se število 9 (binarno 1001) prestavi na izhod števca. Ko se ta podatek prestavi na izhod števca, prikazovalnik tedaj prikazuje število 9, program v krmilniku Alpha poskrbi, da se priključek 11 (Load) prestavi iz

logične 0 v logično 1. Števec sedaj normalno šteje v ritmu števnih impulzov od 9 do 0.

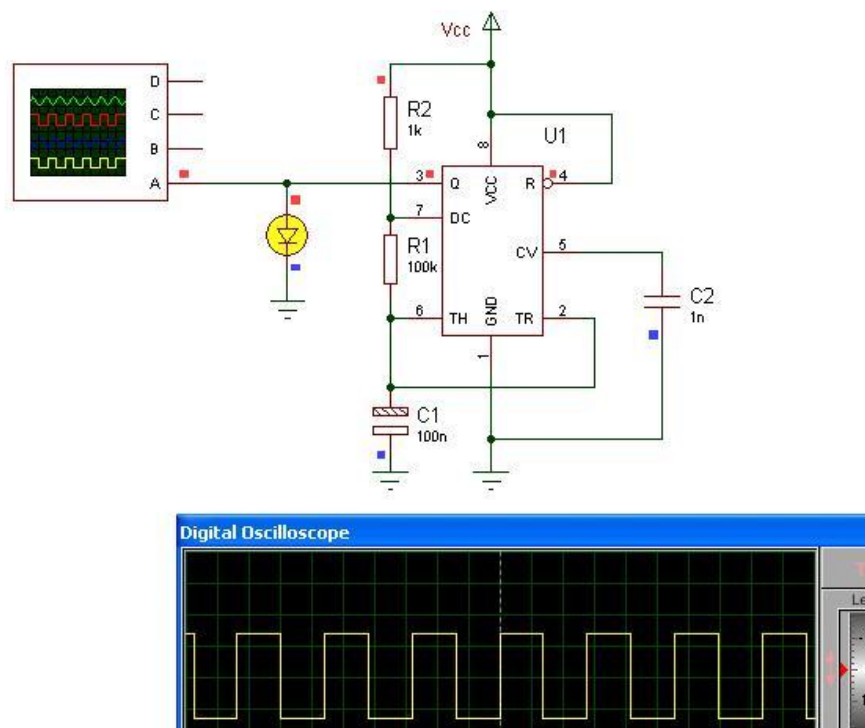
Za pravilno delovanje števca, sva morala upoštevati navodila proizvajalca. Reset vhod mora biti med delovanjem v stanju logične 0 (povezan na maso preko pull-down upora), neizkoriščen števeni vhod, v najinem primeru števeni vhod za štetje gor, pa mora biti v stanju logične 1 (povezan na napetost +5 V).

Števec dobi šteвне impulze iz astabilnega multivibratorja, ki generira impulze s frekvenco 1 Hz.

3.3.4 Časovnik NE 555

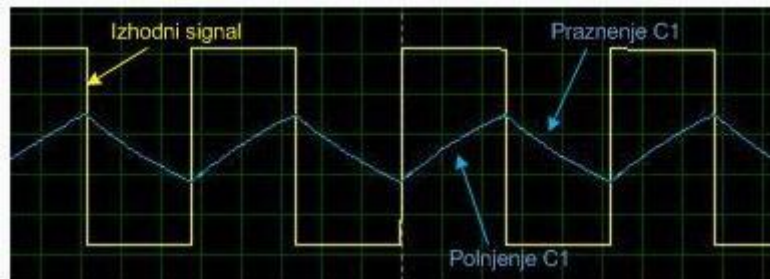
Časovnik NE 555 je integrirano vezje, s katerim lahko generiramo pravokotne impulze s frekvenco, ki je odvisna od uporabljenih elementov vezja.

Ko je vezje priključeno na napetost +5 V, se kondenzator C_1 polni preko obeh uporov (R_2 in R_1). Ko se kondenzator C_1 napolni na $2/3$ napajalne napetosti, komparatorja v časovniku preklopita in kondenzator se začne prazniti preko upora R_1 . Ko vrednost napetosti na kondenzatorju pade na $1/3$ napajalne napetosti, se postopek ponavlja od začetka. V času polnjenja kondenzatorja je na izhodu visoki nivo, v času praznjenja pa nizki (slika 16).



Slika 15: Astabilni multivibrator s časovnikom NE 555.

Impulze, ki jih generira astabilni multivibrator, sva priključila na števeni vhod dekadnega števca. Če je frekvenca teh impulzov 1 Hz, se bo vsako sekundo generiral en impulz. Tukaj se pa večkrat pojavi težava, saj tako natančno frekvenco težko določimo z uporabljenimi elementi vezja.



Slika 16: Napetost na kondenzatorju C1 in izhodni signal astabilnega multivibratorja.

Poskusila sva biti čim bolj natančna in se približati tej vrednosti frekvence. Frekvenco določimo z vrednostjo upora R1 in upora R2 ter vrednostjo kondenzatorja C1. Najlažji način, kako določiti vrednosti teh elementov je ta, da uporabiva spletno stran, kjer je ponujen program za izračun frekvence ob izbranih vrednostih teh elementov.

Slika 15: Program za računanje frekvence za čip 555

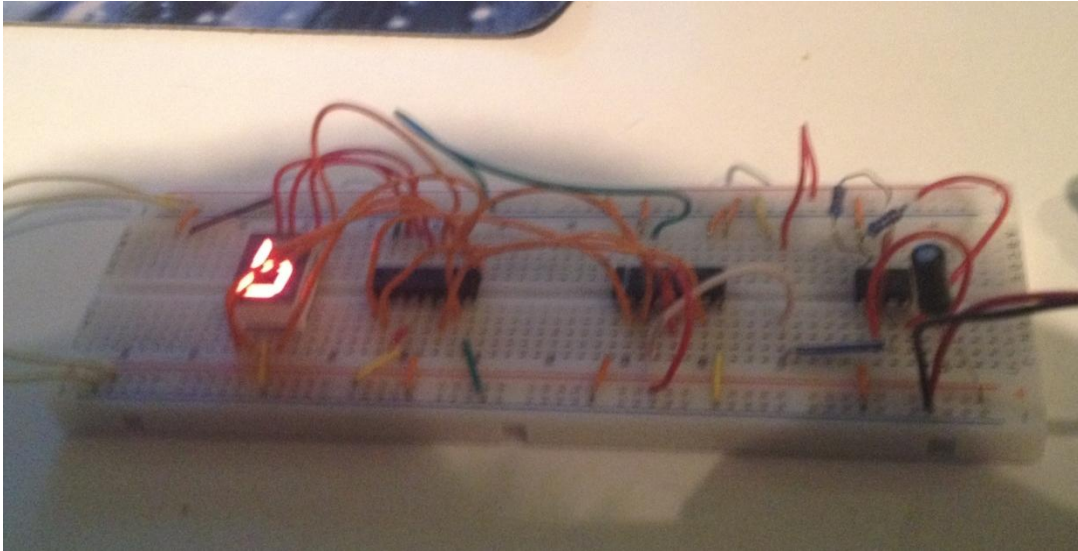
Najbolj sva se približala frekvenci 1 Hz z elementi:

R1 = 1 k Ω ,

R2 = 690 k Ω in

C1 = 1 μ F.

Ko sva izbrala vse elemente, sva začela izdelovati vezje. Pri tem sva morala biti zelo pozorna na priključke integriranih vezij, saj sva se večkrat zmotila. Vezje sva najprej sestavila na eksperimentalno ploščico, ga priključila na napetost in preizkusila delovanje.

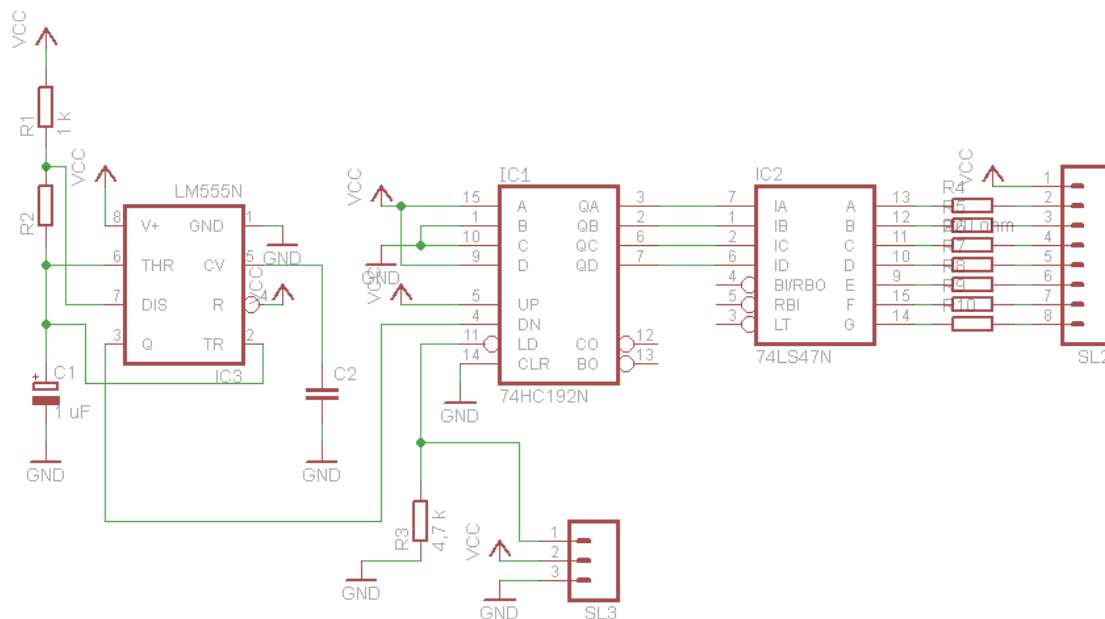


Slika 17: Sestavljeno vezje na eksperimentalni ploščici.

3.4 Izdelovanje tiskanine

Za lepšo predstavitev v maketi sva se odločila, da bova izdelala tiskano vezje. Odločila sva se za uporabo programa Eagle. Najprej sva morala v knjižnici poiskati elemente vezja, ki sva jih potrebovala za izdelavo elektronskega načrta. Uporom sva v knjižnici povečala premere padov, saj so privzeti premajhni. Ko sva vstavila vse potrebne elemente, sva jih morala pravilno povezati. Pri povezavah sva morala biti zelo previdna in upoštevati vezje, ki sva ga že izdelala na eksperimentalni ploščici, saj če bi naredila manjšo napako, vezje ne bi pravilno delovalo.

Števec

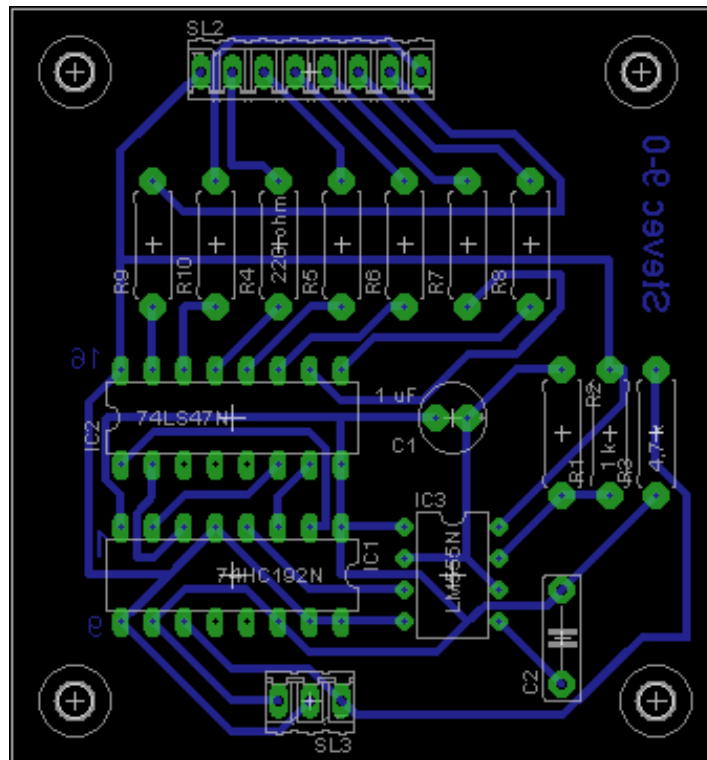


Slika 18: Elektronski načrt generatorja impulzov in števca, sestavljen v okolju Eagle.

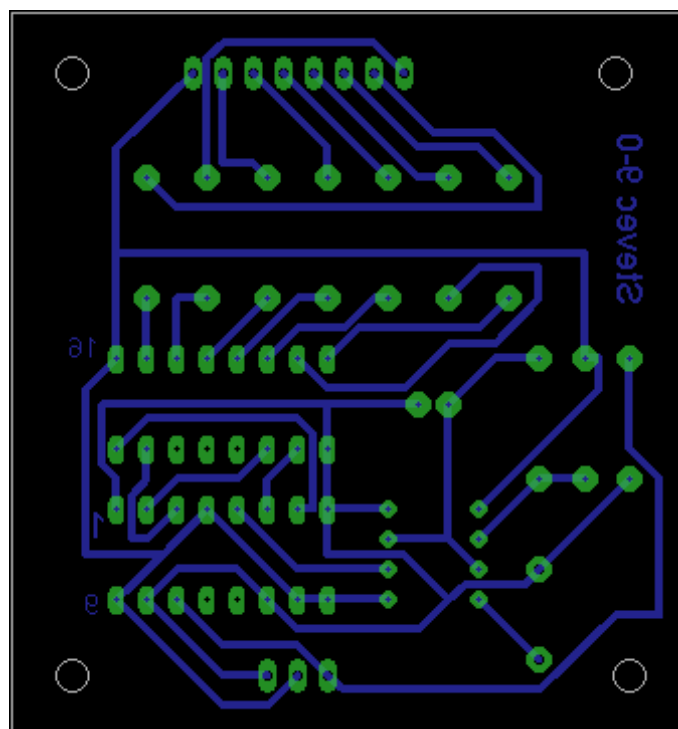
Po izdelavi elektronskega vezja, sva v okolju Eagle izdelala načrt pozicije elementov na tiskani ploščici. Ko sva pozicije elementov na ploščici določila, nama je program izdelal ustrezne povezave. Za 100 % uspešnost povezav sva morala pozicije nekaterih elementov spremeniti oziroma obrniti v drugačen položaj. S tem sva imela kar nekaj težav, na koncu sva le izbrala takšno pozicijo, da je bilo konstruiranje tiskanine 100 % uspešno.

Načrt pozicije elementov na ploščici tiskanega vezja (slika 19) nama je koristil pri vstavljanju in spajkanju elementov vezja v ploščico.

Za izdelavo tiskanine po fotopostopku, pa sva morala odstraniti elemente, napise in oznake, ki jih okolje Eagle prikazuje na zgornji (top) strani ploščice. Nekatero pomembne oznake oziroma napise pa sva dodala na spodnjo stran (bottom) ploščice. To nam koristi, da podnožja, kasneje pa integrirana vezja, pravilno vstavimo in zaspajkamo v tiskano ploščico. Načrt tiskanine sva kopirala na prozorno folijo, ploščico pa preko nje osvetljevala z UV svetlobo okoli 5 minut. Po osvetljevanju sva jo potopila v razvijalec, nakar sva jo zjedkala v jedkalni raztopini.



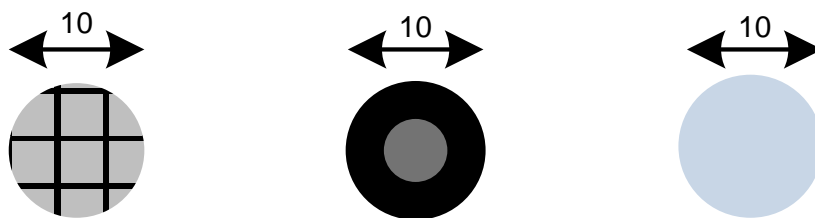
Slika 19: Pozicija elementov na plošči tiskanega vezja.



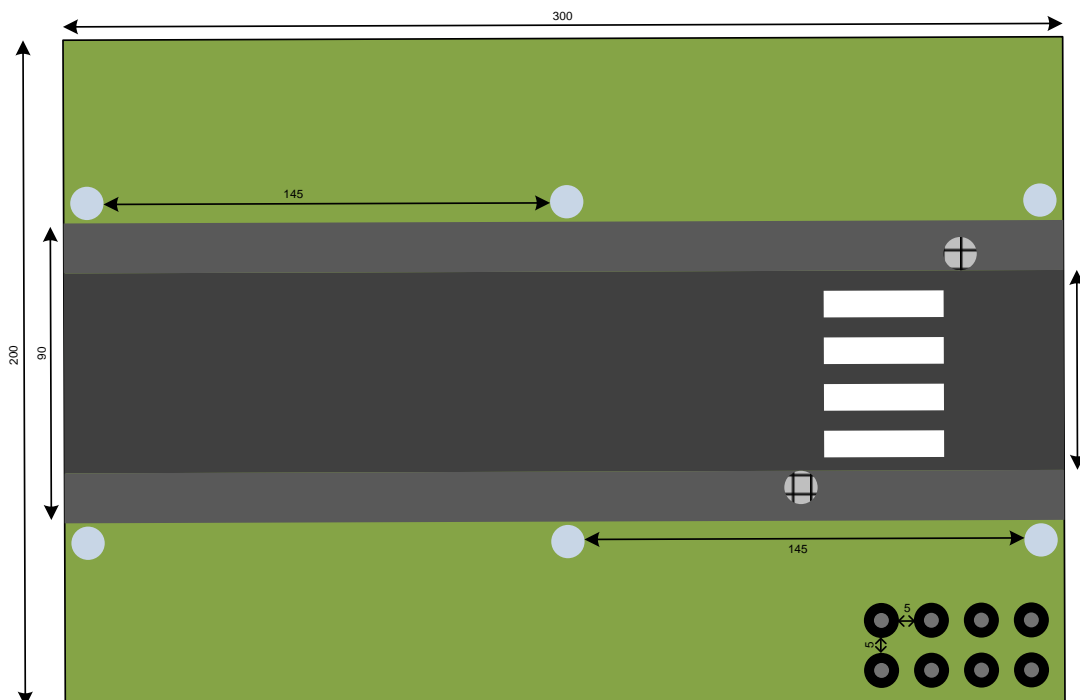
Slika 20: Načrt tiskanine.

3.5 Načrt za izdelovanje makete

Ko sva razmišljala kako bi izdelala maketo, sva se odločila da bova naredila načrt. Odločila sva se, da bova uporabljala program Visio, saj smo se ga naučili uporabljati v šoli. Ko sva začela z izdelovanjem načrta, sva morala najprej izbrati mere za ploščo, kako bo velika. Odločila sva se, da bo dolžina plošče 30 cm, širina pa 20 cm. Plošča bo dovolj velika za lep prikaz najine naloge, hkrati pa ne bo zavzela preveč prostora. Ko sva narisala samo ploščo sva nadaljevala z razporeditvijo elementov, ki jih bo maketa vsebovala. Maketo sva narisala brez kakršnih težav in nato še vstavila vse potrebne mere za natančno izdelovanje makete. Načrt prikazuje slika 21.



Slika 21: Osnovne komponente makete.



Slika 22: Načrt makete.

5 Družbena odgovornost

Ta raziskovalna naloga je bila sprva usmerjena na varčevanje, kako bi lahko pri javni razsvetljavi varčevali. Z uspehom pri tem pa bi seveda vplivala tudi na okolje, saj z varčevanjem z električno energijo ohranjamo okolje čistejše, ker ni tako velike potrebe po energiji in je ni potrebno toliko proizvajati. Tako bi se tudi okolje manj obremenjevalo. Začela sva razmišljati, koliko energije bi se privarčevalo v najinem kraju, kjer imamo 10 luči, ki porabljajo 100 W. Poraba je tako velika, ker v najinem kraju še ne uporabljamo LED svetila, ki so bolj varčna. LED svetila imajo skoraj polovično porabo, 57 W. Tako bi lahko že s prestopom na LED svetila znižali porabo energije. Upoštevamo lahko še čas energije, ki bi jo z opisanim krmiljenjem privarčevali, na dan približno 5 ur, ko bi bila vklopljena samo polovična razsvetljava. Če odbijemo še čase, ko je vključena javna razsvetljava pri dnevni svetlobi, bi na koncu prišli do skoraj 75 % manjše porabe električne energije.

Pri družbeni odgovornosti ne smemo pozabit, da sva na semafor vključila števec in s tem pešce seznanila s podatkom o času, ki ga imajo na voljo, da pridejo čez cesto. S tem bi lahko povečali varnost, saj veliko ljudi, po najinem mnenju, hodi čez cesto počasi, ker mislijo da imajo ogromno časa, s tem pa ogrožajo svojo varnost, ovirajo pa tudi promet.

6 Sklep

Ko sva končala z izdelavi raziskovalne naloge, sva bila zelo olajšana, saj nama je zavzela veliki delež prostega časa. Bila sva tudi zelo zadovoljna z raziskovalno nalogo, saj je vse delovalo in sva se tudi veliko naučila, predvsem programiranje krmilnika Alpha in uporabo programa Eagle za izdelavo tiskanih vezij. Pridobila sva tudi veliko znanja o izdelovanju vezij.

Sedaj, ko sva končala s pisanjem raziskovalne naloge in načrtovanjem, bova pričela z izdelavo makete. Kot sva že podala v sami raziskovalni nalogi, že imava izdelan načrt in imama približno predstavo, kako bi izdelala maketo.

S to raziskovalno nalogo sva se tudi veliko naučila o tem, kako deluje javna razsvetljava in o varčevanju z energijo. Videla sva tudi, kako manjši kraji ter vasi počasneje sledijo energetske varčnemu razvoju. Upava, da sva se s to raziskovalno nalogo veliko naučila in bova lahko naredila še več takšnih projektov.

7 Viri

[1]

[http://www.beijer.no/web/imagebank.nsf/0/36A348F8BDC797B7C1257976004EDD49/\\$file/be_480px_alpha.jpg](http://www.beijer.no/web/imagebank.nsf/0/36A348F8BDC797B7C1257976004EDD49/$file/be_480px_alpha.jpg)

[2]

http://www.shinlight.com/photo/shinlight/editor/20131029143934_19162.jpg

[3]

<http://www.radioaktual.si/?mod=aktualno&action=viewOne&ID=22791>

[4]

<http://nila89.blogspot.com/2010/12/seven-segment-circuit.html>

[5]

<http://forum.allaboutcircuits.com/showthread.php?p=373803>