Zveza za tehnično kulturo Slovenije

58. srečanje mladih raziskovalcev SLOVENIJE 2024

Pametne LED luči

Raziskovalno področje ZOTKS: aplikativni inovacijski predlogi in projekti

(Elektrotehnika, elektronika – inovacijski predlog)

Avtor: Emanuel Pintar Mentor: Branko Potisk Somentor: Mladen Pintar Srednja šola: Srednja elektro-računalniška šola Maribor

Maribor, 2024

KAZALO VSEBINE

K	AZAL	O SL	.IK:i	i
K	\ZAL(0 T <i>i</i>	ABELE: ii	ii
P	OVZE	TEK	:	I
ZÆ	AHVA	LA:	I	I
1	U١	VOD):	1
2	Μ	eto	DOLOGIJA DELA:	3
	2.1	N	lačrtovanje in namen pametne LED razsvetljave:	3
	2.2	Ν	lačrtovanje elektronike:	3
	2.3	N	lačrtovana izvedba:	3
3	lzo	dela	va	1
	3.1	0	hišje pametne LED luči	1
	3.2	E	lektronika in LCD za pametne led luči	5
	3.3	Ρ	rogrami za LCD in pametne led luči:1	5
	3.3	3.1	Program za LCD 4D System:1	5
	3.3	3.2	Program za pametne luči19	Э
	3.3	3.3	Program Microsoft Visual basic RS2322	5
	3.3	3.4	Vmesnik USART LCD na USART luči:	5
	3.3	3.5	Montaža in zagon pametnih led luči:	C
	3.4	Ρ	RIMERJAVA CEN	2
4	Sk	lepi		4
5	DF	ruži	BENA ODGOVORNOST:	5
6	ZA	KLJ	UČEK:	5
7	VI	ri in	N LITERATURA:	7
	7.1	S	pletni viri:	7
	7.2	К	njižni viri:	7

KAZALO SLIK:

Slika 1: 3D model ohišja	5
Slika 2: Gumjasti kalupa za ohišje	5
Slika 3: Odlitki v kalupu	5
Slika 4: Končano ohišje	5
Slika 5: Program CubeMX	6
Slika 6: Shema elektronskega vezja	7
Slika 7: Narisano elektronsko vezje pripravljeno za izdelavo	8
Slika 8: Panel večih elektronskih vezij	8
Slika 9: 3D pogled elektronskega vezja	9
Slika 10: Nanašanje cina na elektronska vezja	10
Slika 11: Elektronska vezja v robotu	10
Slika 12: Delovanje robota	11
Slika 13: Položena elektronska vezja	11
Slika 14: Pečica za spajkanje elektronskih vezij	11
Slika 15: Spajkana (pečena) elektronska vezja	12
Slika 16: Končan panel elektronskih vezij	12
Slika 17: Razlomljena elektronska vezja	12
Slika 18: Slika LED luči z zadnje strani	13
Slika 19: Slika LED luči s sprednje strani	13
Slika 20: Shema vezave na komunikacijo	13
Slika 21: Shema vezave displeja in krmilne elektronike	14
Slika 22: Konektor za komunikacijo Slika 23 Vzava displeja in krmilne elektronike	14
Slika 24: Program 4D systems	16
Slika 25: Izbira našega displeja	17
Slika 26: Izbira programa	17
Slika 27: Oblikovanje prikaza tipk na displeju	18
Slika 28: Končana postavitev tipk na displeju	18
Slika 29: Pini našega mikrokrmilnika za en priključek	20
Slika 30: Pini našega mikrokrmilnika	20
Slika 31: Nastavitev mikrokontrolerja	21
Slika 32: Nastavitev mikrokontrolerja	21
Slika 33: Programiranje komunikacije med displejem in krmilno elektroniko	22
Slika 34: Računalniški terminal	26
Slika 35: Razporeditev led luči v prostoru	
Slika 36: Montiranie led luči	28
Siku 50. Wohlhunje ieu luer	28 30
Slika 37: Kočan izdelek	28 30 30
Slika 37: Kočan izdelek Slika 38: Končan izdelek	28 30 30 31
Slika 37: Kočan izdelek Slika 38: Končan izdelek Slika 39: Končan krmilni displej	28 30 30 31 31
Slika 37: Kočan izdelek Slika 38: Končan izdelek Slika 39: Končan krmilni displej Slika 40: DALI luč	28 30 30 31 31 32
Slika 37: Kočan izdelek Slika 38: Končan izdelek Slika 39: Končan krmilni displej Slika 40: DALI luč Slika 41:DALI Gateway	 28 30 30 31 31 32 32
Slika 37: Kočan izdelek Slika 38: Končan izdelek Slika 39: Končan krmilni displej Slika 40: DALI luč Slika 41:DALI Gateway Slika 42: DALI Gateway	 28 30 30 31 31 32 32 33

KAZALO TABELE:

Гаbela 1	16
Гabela 2	34

POVZETEK:

Kot nalogo smo si zadali izdelati 42 krmiljenih led luči, ki so montirane v strop in so nastavljive preko LCD prikazovalnika na dotik, ki smo ga sami izdelali. Vsaka led luč ima svoj naslov in jo je možno posamično vklapljati in izklapljati ter nastaviti moč. Osnovna ideja je bila preprosta in sicer izdelati plastična vgradna ohišja za pametne led luči, elektronike za njih ter LCD prikazovalnik na dotik, ki bo krmilil led luči. Prvi problem je nastal pri izdelavi ohišja za elektroniko, kjer smo odkrili za nas novo področje in nov izziv in sicer izdelava kalupa in izdelava ohišij iz zalivnih smol. Drugi problem je nastal pri komuniciranju z led lučmi preko USART komunikacije, saj smo morali spustiti hitrost komunikacije zelo nizko zaradi uporabe navadnega kabla. V nalogi smo spoznali izdelavo kalupov in izdelavo ohišij iz zalivnih smol (polimerne mase), postopek industrijske izdelave tiskanih vezij in elektronike z mikrokrmilniki ter programiranje mikrokrmilnikov. Praktični izdelek je z veliko truda lepo uspel in prišli smo do veliko novih spoznanj. To je dragoceno znanje, ki nam bo koristilo pri prihodnjih projektih. Pametne LED luči so inovacijski predlog saj smo v primerjavi z ostalimi izdelki na trgu zelo konkurenčni z ceno ter z enostavno izdelavo.

Ključne besede: mikrokrmilnik, LED luči, izdelava tiskanega vezja, SMD komponente, komunikacijski protokol, RS232

SUMMARY:

We set ourselves the task of producing 42 controlled led lights that are mounted in the ceiling and adjustable via an LCD touch screen that we made ourselves. Each led light has its own address and can be turned on and off individually. The power can be adjusted, too. The basic idea was simple, namely to make plastic built-in cases for smart LED lights, their electronics as well as an LCD touch screen that will control the LED lights. The first problem arose when manufacturing the housings for the electronics. We discovered a new area and a new challenge for us, that was the manufacture of molds and the manufacture of housings from potting resins. Another problem arose when communicating with the LED lights via USART communication, as we had to drop the communication speed very low due to the use of a regular cable. In the researchpaper, we learned about the production of molds and the production of housings from potting resins (polymer mass), the process of industrial production of printed circuits and electronics with microcontrollers, as well as the programming of microcontrollers. We put a lot of efford into the practical product, which was a great success and we came to a lot of new findings. This is valuable knowledge that will benefit us in future projects. Smart LED lights are an innovative proposition, as compared to other products on the market, we are very competitive in terms of price and ease of manufacture.

Keywords: microcontroller, LED lights, printed circuit manufacturing, SMD components, communication protocol, RS232

ZAHVALA:

Zahvaljujem se staršem, ker sem lahko uporabil strop za mojo raziskovalno nalogo. Hvala tudi mentorjema za podporo in pomoč ter nasvete pri izdelovanju praktičnega izdelka. Zahvaljujem se tudi družinskemu podjetju za finančno podporo za ves potrebni material, prav tako se zahvaljujem podjetju, ki je zagotovilo brezplačno izdelavo tiskanih vezij.

1 UVOD:

Cilj naloge je izdelati 42 pametnih led luči, katerim lahko posamično nastavljamo moč oziroma svetilnost ter posamično vklapljamo in izklapljamo vsako luč posebej preko LCD prikazovalnika na dotik. Izziv pa je, da skoraj vse izdelamo sami in glavni cilj ceneje od komercialnih produktov in z enakimi lastnostmi.

Zakaj ravno 42 led luči? Zato, ker se je število 42 lučk izšlo na stropu zraven naše kuhinje. Starši so mi namreč dovolili, da strop opremim z LED lučmi, ki jih v nalogi imenujemo pametne LED luči, ker smo imeli za cilj izdelati ne samo prilagodljivo upravljanje posameznih luči, temveč tudi avtomatično prilagajanja svetilnosti in s tem tudi varčnejšega sistema osvetlitve prostora. V nadaljevanju uporabljam le izraz pametne LED luči.

Strop sva z očetom izdelala iz OSB plošč. Pred izdelavo stropa sem se z očetom dogovoril o montaži lučk, kar je pomenilo počasnejšo montažo. Namreč pri montaži OSB plošč je bilo potrebno vrtanje lukenj za luči in vstavljanje kablov za luči, saj bi kasnejša montaža kablov bila skoraj nemogoča. Tako je nastal strop z 42 luknjami in kabli. Vstavili smo več parov kablov saj še nismo točno vedeli kako bodo priključene pametne LED luči. Mentor mi je dejal, da bodo moje luči obsežen projekt vendar mu nisem verjel, ker sem podcenjeval raziskovalno nalogo, ki se je kasneje spremenila v inovacijski predlog, saj smo uporabili lastni način komunikacije med upravljano enoto in lučmi. Uspelo nam je izdelati pametne LED luči z nizko ceno v primerjavi z konkurenco.

V začetku smo si takole predstavljali potek naloge, ki pa se je bistveno spremenil. Najprej smo imeli željo izdelati ohišja s tiskanjem na 3D tiskalniku. Kar pomeni narisati v CorelCAD ohišje za led luč ter nato to ohišje natisniti s 3D tiskalnikom 42 krat. Uspelo nam je je narisati ohišje v CorelCAD a, ko smo ga še natisnili na 3D tiskalniku je pa to trajalo več kot 4,5 ure in to samo za eno ohišje. Hitro smo ugotovili, da tako ne bo šlo natisniti 42 ohišij. Mentor pove, da obstaja rešitev in sicer kalup in vlivanje ohišij iz zalivnih smol. Tako smo se lotili spoznavanja postopka vlivanja ohišij s kalupi in zalivnimi masami, ki sicer ne spadata v področje elektronike, a nam elektronika ne zadostuje, za zaključitev naloge, če nimamo ohišij.

V drugem delu naloge smo predvideli izdelavo elektronike za upravljanje 12V led lučko, ki bo imela mikrokrmilnik z PWM izhodom za nastavljanje napetosti na lučki in s tem svetilnosti ter UART (serijski) vhod po katerem bo pametna LED luč dobila komande za svetilnost ter vsaka luč bo imela svoj naslov tako, da se luč lahko posamično krmili. Tu smo naletel na več problemov, ki jih nismo predvideli, saj nismo izdelovali le vzorčnega izdelka.

Problem je bil kako izdelati čez 40 kom elektronik za luči. Eno tiskano vezje elektronike ni problem ročno s pinceto naložit SMD komponente, ampak ko je količina večja zna biti to problem. Mentor predlaga izdelavo panelov in polaganje s polagalko (P&P robot), kar pa je zame povsem novo. Prav tako izdelava šablone za polaganje SMD komponent. V tem delu se spoznamo s serijsko proizvodnjo elektronike.

Tretja zadeva, ki je nismo predvideli pa je bila kako nastaviti vsaki lučki svoj naslov. Zopet se je zakompliciralo, ker smo imel več kot 40 lučk. Ni problem v C programu nastaviti naslov (adressa) in generirati kodo ter sprogramirati mikrokrmilnik, če jih je le nekaj. Zato pade odločitev, da bo naslov luči v flash spominu mikrokrmilnika in jo bo možno spreminjati. To pa je pomenilo izdelavo majhnega programa v Microsoft Visual Basic za nastavitev naslova za posamično luč. Za LCD smo uporabili LCD proizvajalca 4D System, ki je pametni LCD na dotik, ki ga poznamo že od prej. Potrebovali smo le še napajanje in vmesnik z mikrokrmilnikom, ki pretvori serijske signale od LCD 4D System v pošiljanje signalov na luči.

Naloga je bila obširna in spoznali smo marsikaj novega, predvsem spoznanje, da je bistvena razlika ali delaš en izdelek ali pa 42 ali več.

2 METODOLOGIJA DELA:

To poglavje smo razdelili na tri podpoglavja in sicer načrtovanje in namen pametne LED razsvetljave, načrtovanje elektronike in montaža.

2.1 Načrtovanje in namen pametne LED razsvetljave:

Želimo doseči ustrezno osvetlitev namenu prostora, estetiko in funkcionalnost našega izdelka. Ker je osnovni cilj izdelati kar se da največ možno v lastni izvedbi in s tem pridobiti ustrezne izkušnje je bil izbran vmesni prostor pri kuhinji v domači hiši. Vseeno pa smo si zadali cilje kot so energijska učinkovitost, prilagodljivost svetlobe, funkcionalnost in na zadnjem mestu tudi estetika. Kriterija izbire luči in obseg glede na krmilni sistem nismo delali ker je izziv prav izgradnja lastnega pametnega sistema in širitve le tega.

2.2 Načrtovanje elektronike:

Na trgu obstaja več standardov za komunikacijo in upravljanje pametne razsvetljave. Omenili bi le najbolj znana in sicer DALI in LON komunikacijo. DALI je standard, ki omogoča digitalno komunikacijo med pametnimi lučmi in krmilniki, LON standard, ki je prav tako namenjen upravljanju zgradb med drugim tudi pametno razsvetljavo omogoča tudi individualni nadzor nad vsako posamezno lučjo v sistemu.

Pametnih nadzornih sistemov za LED osvetlitev, ki omogočajo nadzor in avtomatizacijo tudi s pomočjo pametnih naprav-mobilnih telefonov. Na trgu obstaja zelo veliko obstoječih pametnih sistemov za LED osvetlitev. Vsem pa je skupno visoka cena.

Odločili smo se da bomo izdelali svojega saj ga lahko popolnoma prilagodimo svojim željam in privarčujemo z lastno izvedbo. Pričeli smo z elektroniko, ki pride na vsako posamično LED luč nato pa smo povezali še elektronike za komunikacijo in krmilni displej.

2.3 Načrtovana izvedba:

Strop bo iz OSB plošč. Stop bo spuščen in bo lahko skrili vse potrebne inštalacijske kable.

Nato smo v strop izvrtala 42 lukenj kot smo že prej omenili saj imamo 42 LED luči na stropu. Pri polaganju OSB plošč smo sproti položili vse potrebne kable ter izvrtali luknje za luči. Kasneje smo priklopili LED luči in jih pri vijačili v strop, za na konec pa smo le še priklopili krmilni displej in vse skupaj testirali.

3 IZDELAVA

Kot prvo bomo opisali izdelavo ohišja za pametne led luči z naslovom »Ohišje pametne led luči«. Kot drugo bi opisal dizajniranje in izdelavo elektronike z naslovom »Elektronika in LCD za pametne led luči«. Kot tretje pa bi na kratko opisal programski del, ki ga poimenujem »Programi za LCD in pametne led luči«. Na zadnje pa bi na kratko opisal montažo in izgled luči na stropu poimenovano »Montaža in zagon pametnih led luči«.

3.1 Ohišje pametne LED luči

Najprej smo v CorelCAD narisali želeno obliko ohišja led luči slika 1 spodaj. Nato smo to narisano obliko pretvorili v stl datoteko in s 3D tiskalnikom natisnili eno ohišje. Tiskanje je trajalo malo manj kot 5 ur in še to pri višini nanosa po 0,3mm. Material za tisk je bila črna PLA plastika, ki je cenovno ugodna a bi tiskanje 42 ohišij trajalo predolgo. Potrebno je bilo najti drugo rešitev, zato smo se na predlog mentorja odločili za izdelavo kalupov in vlivanje ohišij kar ni bilo načrtovano ob začetku.

Začetno znanje o izdelavi kalupov je bilo preskromno, zato smo z mentorjem odšla do podjetja SAMSON Kamnik. Tam naju je v trgovini sprejel prijazen gospod, ki je pokazal postopek izdelave silikonskega kalupa. Naše natisnjeno ohišje smo imeli zraven in smo kar tam naredili kalup za luč. V slabi uri smo imeli kalup.

Postopek izdelave kalupa je precej enostaven. Najprej je gospod vzel kos ravne podlage podobne plastiki debeline približno 5mm. Na to podlogo je z dvostranskim lepilnim trakom prilepil naš 3D natisnjen model (isto kot je na Sliki 1). Nato je vzel plastični lonček primerne velikosti in nam pokazal, da mora biti lonček po premeru vsaj 2cm večji od premera mojega ohišja. Lončku je nato odrezal dno in ga poveznil nad naš model tako, da je bilo dno lončka na vrhu. Nato je gospod vzel navaden plastelin in ga nanesel okoli lončka, ki se je stikal podlage. Plastelin bo držal lonček k podlagi in hkrati skrbel, da ne steče silikonska masa iz lončka. Nato je vzel Modrin, ki je silikonski kavčuk za izdelavo kalupov. Modrin vsebuje A in B komponento, ki ju je treba stehtati in zmešati v pravem razmerju. Nato je v najin lonček počasi odlil zmešan Modrin. Sedaj je bilo potrebno le še čakati približno eno uro in že se je Modrin strdil. Tako smo dobili prvi vzorec kalupa, ki je na sliki 2.

Potrebovali smo veliko ohišjih in zato več kalupov. Delo smo nadaljevali doma. Dva kalupa smo naredili še z istim lončkom doma. Nato smo še naredili enak kalup v obliki kocke in sicer tako, da smo namesto lončka okoli mojega modela postavili štiri deščice in jih pritrdili s plastelinom. Tako smo imeli že štiri izdelane kalupe, ki so na sliki 2. Gospod v Samson Kamnik je priporočal zalivno maso KRONOS, ker se zelo hitro strdi in se kasneje lepo obdeluje (brusi, vrta...). Za KRONOS je opozoril, da postane vroč tudi preko 60°C, ko zmešaš A in B komponento in da imamo samo 20s do 40s časa, da jo vlijemo v kalup, ker po 60s postaje že žele. Najbolj smo bili presenečeni nad odgovorom, ko smo ga vprašali, kako dolgo pa je čas, da se strdi KRONOS. Gospod nam je odgovoril, da pri natančnem razmerju in pri debelini sten odlitka do 5mm lahko KRONOS vzamemo iz kalupa po 4 do 6 minutah. Zelo smo se razveselili saj je to pomenilo štiri ohišja na 6 minut, praktično pa 10 minut. To je pomenilo izdelavo vseh

ohišij okoli dveh ur. Pozabili smo še omeniti, da smo dobili ločevalec v obliki spreja s katerim vedno pred odlivanjem posprejamo kalup. Ločevalec se uporablja zato, da se KRONOS lažje odstrani iz kalupa. Odlitki v kalupu so na sliki 3. Na sliki 4 pa je odlitek, ki smo ga še brusili.



Slika 1: 3D model ohišja



Slika 2: Gumjasti kalupa za ohišje



Slika 3: Odlitki v kalupu



Slika 4: Končano ohišje

3.2 Elektronika in LCD za pametne led luči

Elektroniko za led luč smo si zamislili z mikrokrmilniki, ki imajo pulzno širinsko modulacijo (PWM) in serijsko komunikacijo in čim manj nogic. Sprva smo se navdušili nad Microchip mikrokrmilniki serije PIC12F, ki so ustrezali našim pričakovanjem, vendar smo bili žal razočarani, ker jih niso imeli nikjer na zalogi. Na razpolago pa smo imeli zmogljivejše mikrokrmilnike in sicer od proizvajalca STMicroelectronic z oznako STM32G030J6M.

Iz razpoložljivih podatkov smo ugotovili, da so ti mikrokrmilniki ARM 32bitni, ki se programirajo v C ali C++. Programator je imel mentor, sam pa si naložim poizkusno verzijo razvojnega programa za ARM mikrokrmilnike, ki se imenuje IAR ARM.

IAR ARM je profesionalno razvojno orodje za programiranje v C ali C++. Poiskali pa smo še STMCube_MX.

STMCube_MX je program, ki se uporablja za programiranje STM mikrokrmilnikov. S tem programom izberemo in definiramo delovanje mikrokrmilnika v našem primeru STM32G030J6M. Ko izberemo v programu naš mikrokrmilnik se nam odpre ogromno različnih funkcij na vsakem od vhodno izhodnih nogic Primer možnih izbir je na sliki 30. Odločili smo se za pulzno širinsko modulacijo preko RS232 z USART komunikacijo.

Naša pametna luč bo samo poslušala zato potrebujemo samo Rx sprejemni signal. Na sliki 5 je nastavitev našega mikrokrmilnika.



Slika 5: Program CubeMX

Sledil je korak načrtovanja krmilne elektronike za posamezno luč. Imeli smo izbrane funkcije na mikrokrmilniku za pina (nogici) in lotili smo se shematskega načrt slika 6.

Za risanje uporabimo Target3001. Na sliki 6 je tudi popravek načrta namreč R2 smo sprva predvideli 1K Ω , kar za eno ali nekaj lučk drži. Namreč komunikacija deluje na 5V signalih, kar bi pomenilo:

 $(5V - 0.6V(V_{BE}))/1K = 4.4mA$

Se pravi imamo tok 4,4mA pri eni lučki, če pa jih imamo 42 lučk to pomeni 184,8mA za komunikacijo, kar je preveč za 5V regulator, ki se napaja z 12V. Ta 5V regulator je napajan z 12V in je na njem padec napetosti 7V, kar pomeni v našem primeru 1,2936W.

12V-5V=7V

7V * 184,8mA = <u>1,2936W</u>

Zato smo R2 upor, ko je bilo že vse narejeno zamenjali na vseh vezjih lučk z vrednostjo 10K in dobili tok po lučki 440μ A in pri 42 lučeh je tok 18,48mA.

 $(5V - 0.6V(V_{BE}))/10K\Omega = 0.44mA = 440\mu A$

 $440\mu A * 42 = 18,48mA$



Slika 6: Shema elektronskega vezja

Naslednji korak je risanje tiskanine. V Target3001 smo narisali PCB tiskanino, ki jo lahko vidite na slika 7. V našem primeru potrebujemo izdelanih 42 takih tiskanin.



Slika 7: Narisano elektronsko vezje pripravljeno za izdelavo

Target3001 zna sam narediti panel za tiskanino. To je funkcija, ki v primeru potrebe več tiskanim prihrani nekaj dela. Izgled panela na katerem je 20 tiskanin je na sliki 8.



Slika 8: Panel večih elektronskih vezij

V Target3001 pa je tudi funkcija za izdelavo šablone. Šablona se uporablja za nanos spajkalne paste na tiskanino in je iz nerjavečega materiala debeline 120µm. Sedaj, ko smo imeli panel smo datoteke le tega poslali v Gromark d.o.o., kjer izdelujejo tiskanine in datoteko šablone v TMTS d.o.o., kjer izdelujejo šablone. Šablono smo dobili že naslednji dan, tiskanine pa čez 4 dni. Oboji so tako tiskanine kot šablone izdelali zastonj kot pomoč pri izdelavi te naloge. Najbrž je k temu pripomogel mentor.

Sedaj smo imel oboje, ter tudi potrebni material. Mentor mi je odprl program za polagalko SMD komponent, kjer smo z lahkoto napisali program za našo tiskanino. Pravzaprav ni nič pisanja le grafično zlagaš kam pride katera komponenta in to samo za eno tiskanino na panelu. Za ostale tiskanine na panelu mu samo poveš koliko jih je po X osi in koliko jih je po Y osi. Na sliki 9 je izgled programa za polaganje SMD komponent. Programu za polaganje pa povemo tudi centrirne točke, ki so označene z rdečim krogom in križcem. Te točke polagalka preveri s kamero, da lahko točno polaga.



Slika 9: 3D pogled elektronskega vezja

Pričeli smo s pripravami za polaganje. Najprej smo vzel ročno držalo za tiskanino in šablono. Tiskanino smo pritrdili v držalo nato pa na tiskanino zelo točno položili šablono in jo pritrdili z lepilnim trakom, kar je dovolj dobro za izdelavo dveh panelov. Sedaj smo preko šablone potegnili pasto za spajkanje, kar je na sliki 10.



Slika 10: Nanašanje cina na elektronska vezja

Ko smo namazali s pasto za spajkanje, smo dvignili šablono in izpod nje vzeli panel tiskanine. Tega smo previdno nesli do polagalke (robot za polaganje tiskanin) ter panel previdno vstavili v polagalko slika 11.



Slika 11: Elektronska vezja v robotu

Zagnali smo program na polagalki in robot je pričel s polaganjem elementov na panel.

Sprva je preverili centrirne točke s kamero, nato pa pričel polagati SMD komponente, pri čemer je vsako komponento najprej nesel na kamero, jo preveril in preračunal in jo nato položil. To dela zelo hitro zato je bilo težko posneti fotografijo slika 12. Na sliki 13 pa je že položena tiskanina.



Slika 12: Delovanje robota



Slika 13: Položena elektronska vezja

Takšno položeno tiskanino je potrebno postaviti v še en stroj, ki je pečica za spajkanje komponent slika14. Ta pečica ima tekoči trak na katerega postavimo naš panel tiskanine slika15, ki se na eni strani pelje v pečico na drugi strani pečice pa pride ven spajkana tiskanina.



Slika 14: Pečica za spajkanje elektronskih vezij



Slika 15: Spajkana (pečena) elektronska vezja

Ostalo je samo še ročno vstavljanje konektorjev, ki niso odporni na visoke temperature spajkalne postaje, in spajkanje slika16.

Po razlomu panela slika 17 smo dobili izdelana krmilna vezja za vse posamezne LED luči.



Slika 16: Končan panel elektronskih vezij



Slika 17: Razlomljena elektronska vezja

Nato smo pričeli s sestavo led luči v ohišje. Vstavili smo elektroniko na ohišje in jo pritrdilio z lepilom slika18. Zatem pa smo z druge strani vstavili LED luč slika19 in jo na elektroniki pri spajkali.



Slika 18: Slika LED luči z zadnje strani



Slika 19: Slika LED luči s sprednje strani

Po sestavi vseh lučk smo se lotili sestave vmesnika iz RS232 na USART na naših pametnih lučeh.

RS232 je standardna komunikacija iz stacionarnih računalnikov. Pri prenosnih računalnikih pa lahko uporabljamo kar USB vmesnik na RS232.

Po RS232 specifikaciji naj bi bili signali na Tx in Rx -12V in +12V vendar smo zmerili signal z osciloskopom in dobili le +-6V signale. Mentor opozori, da eno je standard drugo pa resnično stanje na portu RS232. Pove tudi, da sodobnejša integrirana vezja lahko dosegajo velike hitrosti tudi pri manjših napetostih. Na sliki 20 je takšen sledilnik, ki ga priključimo na RS232 na drugi strani pa na sprejemni signal Rx na lučki. Naprava samo pretvori pozitivni RS232 signal v pozitivni signal na lučki in negativni RS232 signal pretvori v 0V signal tako, da nimamo negativnih signalov na lučki. Ta vmesnik bomo uporabljali kot nastavitev naslovov na LED lučeh in kot test komunikacije z LED lučmi.



Slika 20: Shema vezave na komunikacijo

Potrebujemo še eden vmesnik in sicer med našim LCD-jem na dotik in LED lučmi. Vmesnik bo pretvoril serijske signale iz LCD-ja v serijske signale za LED luči. Shema vmesnika je na sliki 21.



Slika 21: Shema vezave displeja in krmilne elektronike

Na sliki 22 in sliki 23 sta izdelana vmesnika.



Slika 22: Konektor za komunikacijo



Slika 23 Vzava displeja in krmilne elektronike

3.3 Programi za LCD in pametne led luči:

Programe smo morali razdeliti na več delov in prvi je za LCD na dotik od proizvajalca 4D System, ki je prosto dostopen. To poglavje bi imenovali: »Program za LCD 4D System«. Predstavlja programiranje nastavitev in grafično oblikovanje LCD-ja. Nato smo se lotili programa za mikrokrmilnikov na LED lučeh. To poglavje bi imenovali: »Program za pametne luči«. V tem poglavju bomo opisali delovanje komunikacije in shranjevanje naslova v spomin flash. Za nastavitev naslovov lučk pa potrebujemo program na računalniku, ki bo pošiljal naslov lučki in ta si bo ta naslov shranila v spomin flash. To poglavje bi imenovali: »Program Microsoft Visual Basic RS232«. S tem programom bomo tudi preizkusili delovanje lučk. Nazadnje pa se posvetimo še programu vmesnika med LCD in LED lučmi, ki ga opišemo v poglavju: »Vmesnik USART LCD na USART luči«.

3.3.1 Program za LCD 4D System:

Za izdelavo programa za LCD prikazovalnik smo uporabili program, ki se imenuje Workshop4ide slika 24. V njem smo izdelali kaj bo prikazano na LCD prikazovalniku pa tudi nastavili kaj bo LCD prikazovalnik sporočil elektroniki ob pritisku na določen gumb. Najprej smo v programu odprl nov projekt in izbral LCD prikazovalnik, ki ga bomo uporabljal in sicer GEN4uLCD-43DT slika 25. Nato smo izbrali program, ki ga bomo uporabili za izdelavo projekta. Izbrali smo program imenovan ViSi Genie saj je najenostavnejši za uporabo, ker ni potrebno programirati izgleda tipk in drsnikov saj lahko izberemo, kar v naprej narejen tipke in drsnike, ki jih program ponuja in jih samo spremenimo po naših željah, programsko kodo pa program z generira sam slika 26.

Ko smo se odločili, da bomo luči na stropu razdelili v tri vrste in dve skupine smo se odločili izdelati projekt za LCD prikazovalnik, ki bo vseboval pet stikal za vklop in izklop vsake vrste in skupine luči posebej in drsnik s katerim bi vsaki vrsti in skupini posamično lahko nastavljal svetilnost. Dodali pa smo še pet gumbov, ki služijo zato, da jih pritisnemo in izberemo kateri vrsti oziroma skupini želimo nastavljati svetilnost. Na koncu smo dodali še stikalo za način po meri, ki ga lahko nastavimo in upravljamo z njim vseh 42 luči kakor želimo in jim prav tako nastavljam svetilnost tudi v tem načinu slika 27. Dodali smo še tekst, spremenili barvo stikal, gumbov in drsnika. Ko smo bili zadovoljni z izgledom smo vsakemu gumbu, stikalu in drsniku določili kaj naj se zgodi ob njihovem dotiku. Izgled na sliki 28. Vse smo nastavili na (report message) kar pomeni, da LCD prikazovalnik pošlje sporočilo kdaj je bila neka tipka, stikalo oziroma drsnik pritisnjen. Na koncu smo program samo še naložili na SIM kartico in jo vstavili v LCD prikazovalnik in preverili, če LCD prikazovalnik pravilno deluje in pravilno pošilja sporočila.

Sporočila, ki jih pošilja LCD so na serijskem portu in sicer 9600bps 8,N,1. 8 pomeni 8 bitni prenos, N pomeni parnost NE (Parity None), 1 pa pomeni 1stop bit. LCD po serijskem portu pošilja vedno šest 8 bitnih števil in sicer števila od 0 do 255. V tabeli 1 spodaj so podatki kaj pošlje LCD ob pritisnjeni tipki na LCD-ju. Vsak tip objekt ima svojo številko in zaporedno številko objekta. Stikala imajo številko 0x3A za tip in zaporedno število od 0 do 5, ker imam 6 stikal. Isto je za tipke in drsnik.

Funkcija	Komanda	Tip	Št. objekta	Vrednost	Vrednost	XOR
	1 Byte	objekta	3 Byte	Н	L	6 Byte
		2 Byte		4 Byte	5 Byte	
Stikalo On/Off Off	0x07	0x3A	0 do 0x05	0x00	0x00	xor
Stikalo On/Off On	0x07	0x3A	0 do 0x05	0x00	0x01	xor
Tipka svetilnost	0x07	0x06	0 do 0x05	0x00	0x00	xor
Drsnik	0x07	0x04	0x00	0x00	0-105	xor
					0-0x69	

Tabela 1



Slika 24: Program 4D systems



Slika 25: Izbira našega displeja



Slika 26: Izbira programa

File Home View Tools Comms Project New Open Save Save File Save Save <t< th=""><th>Buttons Digits Gauges I/O</th><th>Inputs Int/Inh Inputs Labels Primi</th><th>tives System/Media</th></t<>	Buttons Digits Gauges I/O	Inputs Int/Inh Inputs Labels Primi	tives System/Media
Form0 Vrsta_1 Vrsta_2 or r or o	Vrsta_3) eer ivetiinost Gin po meri) eer Svetiinost	Object Inspector Form 0 Object Form0 Properties Events Property Value Name Form0 Alas Form0 Batype Color Color BACK Image (Vone) If Source	

Slika 27: Oblikovanje prikaza tipk na displeju

Form0				8
	Vrsta_1	Vrsta_2	Vrsta_3	
	Custilnost	Svetilpost	Svetilpost	
	Svetimost	Svetimost	Svetimost	
	Skupina_1	Skupina_2	Način po meri	
	Guildhaut	Gunkilmank	Custilanat	
	Svetilhost	Svetilnost	Svetilnost	
			-	

Slika 28: Končana postavitev tipk na displeju

3.3.2 Program za pametne luči

V tem delu bomo na kratko opisali program za STM32G030J6M mikrokrmilnik, ki krmili preko FET tranzistorja led luč. Prav tako pa sprejema serijske signale komand. Najprej sem odprl program STM32CubeMX, ki je program za nastavitev mikrokrmilnikov. Izbrali smo naš mikrokrmilnik STM32G030J6M, ki ima samo 8 priključnih nogic od tega so 2 za napajanje. Ko smo pričeli z nastavitvami in kliku na eno izmed priključnih nogic se odpre ogromno funkcij, ki so lahko na izbrani nogici, kar se vidi na sliki 29 in 30. Začetnika to malce ustraši, a se je potrebno osredotočiti samo na to kar želi nastaviti. V našem primeru samo na USART Rx in na časovnik za PWM vse ostalo pa naj pustiti pri miru. Prav tako nismo spreminjali časovnikov mikrokrmilnika (Clock configuration) saj jih program sam nastavi.

Izbrali smo USART1 in nastavili samo kar je označeno rumeno na sliki 31. Nastavili smo »Single wire« ter nato še 115200,8,N,1 (115200 Bits/s, 8 bitni prenos, Parity = None, 1 Stop Bit). Po posvetu smo znižali hitrost prenosa podatkov in sicer zaradi predhodno pripravljenih kablov, ki jih uporabljamo za napajanje in za prenos podatkov. Kabel je navadni 3x0,75mm² z modro, rjavo in rumeno zeleno žico. Modro bomo uporabili kot minus, rjavo kot +12V in rumeno zeleno za prenos podatkov. Mentor opozori, da se načeloma rumeno zelena uporablja samo za ozemljitev in da ni ravno prav jo uporabiti za komunikacijo. A ker dodatnih vodnikov nimamo, ozemljitve nismo načrtovali zaradi 12V napajanja, bo tako kar v redu. Mentor pove, da takšen, uporabljen kabel ima notranjo kapacitivnost in induktivnost in ni namenjen komunikaciji, ter svetuje čim nižji prenos podatkov in sicer 1200Bits/s.

Tako nastavimo prenos na 1200Bits/s. Nastavili pa smo še Interupt za USAR1. To potrebujemo za sprejem podatkov.

Nato smo nastavili še časovnik TIM1 in PWM kar je razvidno iz slike 32. TIM1 smo nastavili na »Internal Clock« notranji uro oziroma oscilator, ki je 16MHz. Nato smo nastavili štetje števca na 100 (»Counter Period (AutoReload Register - 16 bits value«), ker želimo nastavljati PWM od 0 do 100%.

Nastaviti je potrebno še frekvenco PWM, ki smo jo najprej nastavili na 50HZ. Delitelj (»Prescaler«) frekvenco dobim takole:

(16MHz / 100(Caunter Period)) / 50Hz = 3200

Pri frekvenci PWM 50 Hz smo ugotovili, da naše oko še zazna rahlo utripanje led luči, če PWM zmanjšamo pod 60%. Zato smo PWM frekvenco dvignili na 60Hz ter na novo izračunal delitelj (»Prescaler«).

(16MHz / 100(Caunter Period)) / 60Hz = 2666,66 = <u>2667</u>

Sedaj smo samo kliknili »GENERATE CODE« in program sam naredi kodo v C za program IAR ARM, ki se uporablja za C programiranje mikrokrmilnikov slika 33.

PC14	0	PA14-	PA14-BOOT0	PA15	PB5	PB6
VDD/ VSS/		PA13 PA12 .	Reset_State ADC1_IN18 SYS_SWCLK USART2_TX GPI0_Input	Reset_State I2S1_WS SPI1_NSS USART2_RX GPI0_Input	Reset_State I2C1_SMBA I2S1_SD SPI1_MOSI SYS_WKUP6	Reset_State I2C1_SCL SPI2_MISO TIM16_CH1N TIM1_CH3
NRST		PB0	GPIO_Output GPIO_Analog EVENTOUT GPIO_EXTI14	GPIO_Output GPIO_Analog EVENTOUT GPIO_EXTI15	TIM16_BK TIM3_CH2 GPIO_Input GPIO_Output	USART1_TX GPIO_Input GPIO_Output GPIO_Analog
		-			GPIO_Analog GPIO_EXTI5	EVENTOUT GPIO_EXTI6

Slika 29: Pini našega mikrokrmilnika za en priključek

USART1_F	₹x		
PB0 Reset State	PB1 Reset State	PA8 Reset State	PA11 [PA9] Reset State
ADC1 IN8	ADC1 IN9	RCC MCO	ADC1 EXTI11
12S1_WS	TIM14_CH1	SPI2 NSS	ADC1_IN15
SPI1_NSS	TIM1_CH3N	TIM1_CH1	12C2_SCL
TIM1_CH2N	TIM3_CH4	GPIO_Input	I2S1_MCK
TIM3_CH3	GPIO_Input	GPIO_Output	SPI1_MISO
GPIO_Input	GPIO_Output	GPIO_Analog	TIM1_BK2
GPIO_Output	GPIO_Analog	EVENTOUT	TIM1_CH4
GPIO_Analog	EVENTOUT	GPIO_EXTI8	USART1_CTS
GPIO_EXTI0	GPIO_EXTI1		USART1_NSS
			GPIO_Input
			GPIO_Output
			GPIO_Analog
	USART1_F PB0 Reset_State ADC1_IN8 I2S1_WS SPI1_NSS TIM1_CH2N TIM3_CH3 GPI0_Input GPI0_Output GPI0_Analog GPI0_EXTI0	USART1_Rx PB0 PB1 Reset_State ADC1_IN8 ADC1_IN9 I2S1_WS TIM1_CH3N TIM1_CH2N TIM3_CH4 TIM3_CH3 GPIO_Input GPIO_Input GPIO_Output GPIO_Output GPIO_Output GPIO_Analog GPIO_Analog GPIO_EXTI0 GPIO_EXTI1	USART1_RX TIM1_CH1 PB0 PB1 PA8 Reset_State ADC1_IN8 ADC1_IN9 I2S1_WS TIM14_CH1 SPI2_NSS SPI1_NSS TIM1_CH3N TIM1_CH2N TIM3_CH4 GPI0_Input GPI0_Input GPI0_Output GPI0_Input GPI0_Output GPI0_Output GPI0_Analog EVENTOUT GPI0_Analog EVENTOUT GPI0_EXTI0 GPI0_EXTI1

Slika 30: Pini našega mikrokrmilnika

	~ @		USART1 Mode and Configuration	
ategories A->	Z		Mode	
System Core	>	Mode Single Wire (Half-Duplex)		~
		Hardware Flow Control (RS232) Disable		~
Analog	>	Hardware Flow Control (RS485)		
Timers	>	Slave Select(NSS) Management Disable		\sim
Connectivity	~			
•			Configuration	
Ø 12C1		Baset Configuration		
Ø 12C2		Reset Comiguration		
		Parameter Settings Ø User Constants	s 📀 NVIC Settings 🔮 DMA Settings 🔮 GPIO Settin	igs
A SPI2		Configure the below parameters :		
USART1		Q Search (Ctrl+F) (0) (0)		6
A USART2		A Basic Parameters		
		Baud Date	1200 Bite/e	
		Word Length	8 Bits (including Parity)	
Multimedia	>	Parity	None	
		Stop Bits	1	
Computing	>	Advanced Parameters		
		Data Direction	Receive Only	
Viddleware	>	Over Sampling	16 Samples	
		Single Sample	Disable	
		ClockPrescaler	1	
		Fifo Mode	Disable	
		Txfifo Threshold	1 eighth full configuration	
		Rxfifo Threshold	1 eighth full configuration	

Slika 31: Nastavitev mikrokontrolerja



Slika 32: Nastavitev mikrokontrolerja



Slika 33: Programiranje komunikacije med displejem in krmilno elektroniko

Za nastavitev PWM smo uporabljali ukaz TIM1->CCR1= od 0 do 100, kar pomeni od 0 do 100%. Ko smo v programu spreminjali vrednosti se je lepo spreminjala svetilnost luči.

Sedaj smo si samo zamislili sprejemni signal, ki bo iz treh (Bytov) 8 bitnih podatkov. Prvi 8 bitni podatek bo naslov luči. Naslov 0 naj se prične pi številu 60 (0x3C) nato 1 pri 61 (0x3D) in tako naprej. Se pravi številka luči + 60 je njen naslov. Drugi 8 bitni podatek pa je moč oziroma svetilnost od 0 do 100% v desetih korakih. Se pravi od 0 do 10 dobimo vrednost od 0 do 100%. tukaj smo izbrali kar karakterje števil od 0 do 9 in carakter ':' za 10. Tretji 8 bitni podatek pa je zaključek komande in sicer karakter 13 (0x0D). Spodaj je nekaj primerov za sprejem komand.

Koda komande:	pomen:
61 48 13	luč številka 1 moč 0 0%
61 43 13	luč številka 1 moč 5 50%
61 58 13	luč številka 1 moč 10 100%
65 48 13	luč številka 5 moč 0 0%
65 43 13	luč številka 5 moč 5 50%
65 58 13	luč številka 5 moč 10 100%

Del podprograma za sprejem na serijskem portu izgleda takole:

```
/*------

Sprejem_CMD_Rx

void Sprejem_CMD_Rx (void)

{

u8 Sprejem_OK;

Sprejem_OK=HAL_UART_Receive_IT(&huart1, (uint8_t *)RxBuffer, RXBUFFERSIZE);

if(Sprejem_OK == 0)

{

if((RxBuffer[2] == 0x0d) && (RxBuffer[0] == Luc_ADDR))

{

Power_Led=10*(RxBuffer[1]-0x30);

TIM1->CCR1=Power_Led;

}

}
```

HAL_UART_Receive_IT je ukaz iz knjižnice, ki ga program sam ustvari. Mi ga le kličemo in mu zapišemo kam naj naloži sprejete podatke »RxBuffer« in naj bo dolžina teh podatkov 3 »RXBUFFERSIZE«, saj želim sprejemati tri 8 bitne podatke. Nato samo preverim ali je zaključni karakter pravi in če je naslov »Luc_ADDR« enak sprejetemu naslovu.

V kolikor je naslov enak potem nastavimo moč luči z PWM. Program smo preizkusili, kar preko terminala RS232 in našega vmesnika za RS232.

Nato smo morali napisati še program za shranjevanje naslova luči.

Odločili smo se, da omejimo naslove lučk na maksimalno 64, kar v našem primeru pomeni 64+60=124. Prvi 8 bitni podatek bo od vrednosti 60 do 124 in pomeni naslove luči. V kolikor pa prvi 8 bitni podatek pride večji kot 124 v našem primeru 165 (0xA5) pa pomeni, da nastavljamo naslov luči in naj si ga pametna LED luč zapomni. V primeru, da je prvi podatek 165 (0xA5) je drugi podatek novi naslov luči ter tretji podatek potrditev 13 (0x0D). Spodaj je prikazan podatek za nastavitev novega naslova.

165 61 13 sprejel ukaz (165) novi naslov je 1 (61-60=1)

0xA5 0x3D 0x0D

V programu smo postavili označbo (zastavica »Flag«), če je po serijskem portu prišel ukaz novi naslov (165 = 0xA5). To označbo smo imenovali »Nova_Addr_F«.

```
if(RxBuffer[0] == 0xA5)
{
    SET_BIT(DFF_reg,Nova_Addr_F);
    Addr_Reg=RxBuffer[1];
}
```

V podprogramu »Nova_Addr_Vpis« pa samo preverjamo ali smo dobili ukaz za novi naslov. Preverjam bit »Nova_Addr_F«.

```
void Nova_Addr_Vpis (void)
{
    if(DFF_reg & Nova_Addr_F)
    {
        Vpis_Flash_Nova_ADDR();
        CLEAR_BIT(DFF_reg,Nova_Addr_F);
        SET_BIT(DFF_reg,DFF_Led_utrip_F); /* za utrip lučke po vpisu novega naslova */
    }
}
```

V kolikor je bil postavljen bit »Nova_Addr_F« se zažene podprogram »Vpis_Flash_Nova_ADDR« s katerim vpišemo v pomnilnik Flash novi naslov. Pri uporabi pomnilnika mikrokrmilnika opozorimo na lastnost, da je pomnilnik sestavljen iz strani (»Page«) od katerih ena stran zajema 2048 osem bitnih vrednosti. Če želimo vpisovati v neko stran moram najprej odkleniti pomnilnik nato zbrisati celo stran vseh 2048 osem bitnih vrednosti ter nato vpisati želeno vrednost. Po vpisu pa zakleniti spomin. Vse skupaj je že pripravljeno v knjižnici za »Flash«.

Tukaj je primer:

```
void Vpis_Flash_Nova_ADDR (void)
 /* Unlock the Flash to enable the flash control register access *********/
HAL_FLASH_Unlock();
 /* Fill EraseInit structure*/
EraseInitStruct.TypeErase = FLASH_TYPEERASE_PAGES;
 EraseInitStruct.Page
                       = 15:
EraseInitStruct.NbPages = 1;
/* Note: If an erase operation in Flash memory also concerns data in the data or instruction cache,
  you have to make sure that these data are rewritten before they are accessed during code
  execution. If this cannot be done safely, it is recommended to flush the caches by setting the
  DCRST and ICRST bits in the FLASH_CR register. */
 if (HAL_FLASHEx_Erase(&EraseInitStruct, &PageError) != HAL_OK)
 {
   Error occurred while page erase.
   User can add here some code to deal with this error.
   PageError will contain the faulty page and then to know the code error on this page,
   user can call function 'HAL_FLASH_GetError()'
  */
 /* Program the user Flash area word by word
  (area defined by FLASH_USER_START_ADDR and FLASH_USER_END_ADDR) ********/
   Address = FLASH_USER_START_ADDR;
                          /* Tole tukaj je moj novi naslov */
   DATA_64=Addr_Reg;
  HAL FLASH Program(FLASH TYPEPROGRAM DOUBLEWORD, Address, DATA 64);
```

```
/* Lock the Flash to disable the flash control register access (recommended
to protect the FLASH memory against possible unwanted operation) *******/
HAL_FLASH_Lock();
```

Da smo vedeli, da je luč dobila novi naslov smo dodali še podprogram, da luč utripne, ko je vpisan novi naslov v luč. Ta podprogram smo imenovali »Led_Utrip«.

```
void Led_utrip (void)
{
    if(DFF_reg & DFF_Led_utrip_F)
    {
        st_Led_utrip++;
        if(st_Led_utrip == 1)
        {
            TIM1->CCR1=80;
        }
        else if(st_Led_utrip >= 500)
        {
            TIM1->CCR1=0;
            st_Led_utrip=0;
            CLEAR_BIT(DFF_reg,DFF_Led_utrip_F);
        }
    }
}
```

Tako smo zaključili program za mikrokrmilnik za luči in se naučili nekaj o C programiranju in knjižnicah.

3.3.3 Program Microsoft Visual basic RS232

Izbrali smo Microsoft Visual basic, kjer smo izdelali program za nastavljanje naslovov luči, pa tudi hkratno testiranje delovanja pametnih LED luči, saj jim lahko preko računalnika in tega programa, ki smo ga izdelali nastavljamo svetilnost in izbiro, če svetijo ter preizkusimo, če imajo pravi naslov.

Naredili smo sledeče. V program vstavili tri NumericUpDown in jim določili imena, ter določili velikost področja do katerega delujejo. Prvima dvema smo nastavili področje delovanja na 0-64 tretjemu, ki pa smo ga uporabili za določanje oziroma nastavljanje svetilnosti LED luči pa področje delovanja od 0-10. Naslove smo poimenovali ADDR krajše za angleško address. Nato smo vstavili še tri gumbe, vsak gumb pa je sodeloval z vsakim od NumericUpDown. S prvim NumericUpDown in prvim gumbom smo vsaki od 42 LED luči preko računalnika nastavili svoj naslov, drugi NumericUpDown smo uporabili pri testiranju svojih luči, kjer smo izbrali LED luči in njen naslov s tretjim NumericUpDown pa smo nastavili moč svetilnosti pri testiranju svojih luči, da smo videli, če vse pravilno delujejo. Nato smo napisali za vse to še programsko kodo. Vsaki LED luči, ki jo priklopimo na računalnik najprej določimo naslov in jo preizkusimo. Program prvo pošlje podatek kode 165 oziroma 0xA5, ki luči pove, da je to komanda za novi naslov, ki je poslan v drugem podatku (60+novi naslov) in v tretjem potrditev 0x0D ali 13. Program je napisan tako, da z NumericUpDown izbiramo novi naslov in mu prištejemo 60, tako dobimo število za novi naslov.

Na koncu pa smo vse LED luči samo še testirali s pomočjo tega programa, ki smo ga izdelali, Test vsebuje preverjanje naslovov pametnih LED luči in pravilnost njihove svetilnosti, da vidimo če se moč svetilnosti pravilno spreminja. Program na sliki 34.

			-	×
Comm. Port Comm. Port OFF Help		Nova ADDR		
ADDR 0	Moc 0	Test ADDR / Moc		

Slika 34: Računalniški terminal

3.3.4 Vmesnik USART LCD na USART luči:

Na sliki 21 je shema vmesnika med LCD in lučmi. Za vmesnik smo vzeli že izdelano razvojno tiskanino in sicer »STM32LVDISCOVERY BOARD«, ki vsebuje mikrokrmilnik »STM32F100RBTx«. V programu STM32CubeMX smo nastavili USART1 za oddajo na luči ter USART2 za sprejem podatkov iz LCD prikazovalnika. USART1 smo nastavili na 1200,8,N,1 in USART2 na 9600,8,N,1 saj LCD prikazovalnik pošilja podatke z 9600Bits/s. Nato smo v programu generirali kodo v C za IAR ARM program. Vmesnik sprejema podatke iz LCD prikazovalnika, ki so v tabeli 1. Vmesnik nič ne pošilja LCD prikazovalniku. Signala Rx in Tx smo predvideli zato, ker bi lahko tudi pošiljali podatke na LCD, mi pa jih trenutno samo beremo. Pred izdelavo programa za LCD prikazovalnik, ki je opisan v poglavju »Program za LCD 4D System« smo si zamislili kako bomo razdelili luči v prostoru slika 35. Na sliki 35 so označene luči in njihovi naslovi od 1 do 42. Cel strop z 42 lučmi zajema dva prostora zato smo večji prostor (prostor 1) razdelili v tri vrste in skupino 12 lučk (skupina 1). Manjši prostor (prostor 2) pa smo poimenovali skupina 2. Razdelitev luči je pomembna saj bomo vsaki vrsti ali skupini luči pošiljali posamično podatke za delovanje.

V IAR ARM smo napisali program, ki sprejema podatke iz LCD prikazovalnika in si postavi zastavice za sprejete informacije. Spodaj je program za sprejem z LCD prikazovalnika. Iz knjižnice smo uporabili funkcijo »HAL_UART_Receive_IT«, ki sama poskrbi za sprejem podatkov mi smo le nastavili naj sprejema 6 podatkov in naj jih shrani v prej definirani pomnilnik.

Spodaj je le delček omenjenega programa.

```
Preveri_Sprejem_Od_4D_USART2
void Preveri_Sprejem_Od_4D_USART2 (void)
{
u8 Sprejem_OK;
    if(st_ob_vklopu_USART2 <3000) //3000 = 3sek
                           //ob vklopu preve 3sek ne preverja ser. porta
    {
    st_ob_vklopu_USART2++;
    return;
    }
    Sprejem_OK=HAL_UART_Receive_IT(&huart2, (uint8_t *)RxBuffer_2, RXBUFFERSIZE);
//LCD poslal vrednost drsnika
    if((Sprejem_OK == 0) \&\& (RxBuffer_2[0] == 0x07))
     if((RxBuffer_2[1] == 0x04) && (RxBuffer_2[2] == 0x00)) //Slider
     {
        Drsnik_Vrednost=RxBuffer_2[4];
        Drsnik_Vrednost_0_do_10=Drsnik_Vrednost/10;
        SET_BIT(DFF_reg,DFF_Sprejel_Drsnik_F);
     }
//LCD poslal vrednost stikal
    if(RxBuffer_2[1] == 0x3A) //LCD poslal vrednost stikala
     {
        switch (RxBuffer_2[2]) //vrednost skilal na LCD od 0 do 5
    {
     case 0x00:
            if(RxBuffer_2[4] !=0)
             SET_BIT(Stikala_reg,Stikalo_0_F);
             CLEAR_BIT(DFF_reg,DFF_Stikalo_0_On_F);
            }
            else
             CLEAR_BIT(Stikala_reg,Stikalo_0_F);
             CLEAR_BIT(DFF_reg,DFF_Stikalo_0_Off_F);
     break;
```



Slika 35: Razporeditev led luči v prostoru

Ko je program za sprejem deloval, smo napisali še program za oddajo serijskega signala na luči. Pri tem pod programu pa gledamo postavljene zastavice iz zgoraj omenjenega programa in lučem pošiljamo podatke. Koliko lučem pošiljamo podatke pa je odvisno od izbrane vrste ali skupine. V kolikor na primer izberem vrsto 1 bomo poslali komande samo lučem od 1 do 8, če izberemo npr. vrsto 3 pa lučem od 17 do 24. Prav tako za skupini. Se pravi gledamo pritisnjeno tipko ali stikalo in glede na vrsto ali skupino pošljem komande določenim pametnim LED lučem. Spodaj je kratek povzetek programa za oddajo na luči. Tukaj je le delček omenjenega programa.

```
/*------
Preveri_Stikala
Preveri_Stikala (void)
{
//Stikalo 0
if(Stikala_reg & Stikalo_0_F)
{
CLEAR_BIT(DFF_reg, DFF_Stikalo_0_Off_F);
if(!(DFF_reg & DFF_Stikalo_0_On_F))
{
// Test_Stikala_On_Off_USART1(0,1);
Stikala_On_Off_USART1(0,1);
HAL_UART_Transmit_IT(&huart1, (uint8_t *)TxBuffer_1,TXBuffer_1_Size);
SET_BIT(DFF_reg, DFF_Stikalo_0_On_F);
}
```

```
}
else
{
    CLEAR_BIT(DFF_reg, DFF_Stikalo_0_On_F);
    if(!(DFF_reg & DFF_Stikalo_0_Off_F))
    {
        Test_Stikala_On_Off_USART1(0,0);
        Stikala_On_Off_USART1(0,0);
        HAL_UART_Transmit_IT(&huart1, (uint8_t *)TxBuffer_1,TXBuffer_1_Size);
        SET_BIT(DFF_reg, DFF_Stikalo_0_Off_F);
    }
}
```

```
//Stikalo 1,2,3,4,5 od tu naprej isto le za druga stikala
```

3.3.5 Montaža in zagon pametnih led luči:

Tukaj bi le na kratko omenili, da je bilo kar nekaj dela preden smo zmontirali 42 LED luči. Primer montaže luči je na sliki 36. Na sliki 37 in 38 pa so vključene LED luči z dveh zornih kotov. Na sliki 39 pa je še LCD prikazovalnik na dotik. Končno je prišel trenutek prvega zagona. Kljub pravilni priključitvi pametnih LED luči niso delovale. Pomoč je ponudil mentor in napovedal, da imamo nekje signalno žico povezano na minus. Vse luči smo pregledali in ugotovili, da je predvidevanje bilo pravilno, ena žička od minusa se je dotikala signalne žice. Ponovno vključimo in vse je pričelo odlično delovati. Navdušen je bil mlajši bratec in tudi starša.



Slika 36: Montiranje led luči



Slika 37: Kočan izdelek



Slika 38: Končan izdelek



Slika 39: Končan krmilni displej

3.4 PRIMERJAVA CEN

Že obstoječi izdelki na tržišču so zelo dragi in stanejo po par sto eurov za kakšne večje projekte pa tudi nad tisoč eurov. Največji konkurenti našemu izdelku so izdelki od proizvajalcev, ki uporabljajo LON in DALI.

LON je tehnologija, temelji na protokolu LonTalk® (ISO/IEC 14908-1), ki deluje preko različnih fizičnih medijev, kot so parični kabli, optična vlakna in brezžične povezave.

DALI (Digital Addressable Lighting Interface) je industrijski standard, ki določa protokol za digitalno komunikacijo in nadzor razsvetljave.

Spodaj je nekaj izdelkov in cen:

Item-No.: LDV P 075144668 *243,66 excl. VAT. excl. shipping costs **** Special item please note deliverytime
I Quantify + add to cart Mark to compare add to list Create new list + add
LEDVANCE

Slika 40: DALI luč

SE MTN6725-0003 KNX DAL	NOUL VE WILL HERINAL	64 Item-No.: SE MTNG SOUTH Controlled (tem-No.: SE MTNG SOUTH Controlled (tem-No.: SE MTNG Controlled (tem-No.: SE MTNG (tem-No.: SE MTNG	SWICHDEININ EXMINISIO 3725-0003 3-5 business days + add to cart re + add	Actuator rg, 5 uniaty
Chern Control Contr			Schneider Electric	

Slika 41:DALI Gateway

HMIP P-DRG-DALI DALI Gateway	Item-No.: HMIP P-DRG-DALI 8252_06
CO3 Com A B HenP-P-DRG-DAL	excl. VAT. excl. shipping costs
<i>्</i> 📰 📡 🛋 👀	home matic 🖻



Tukaj pa so še nekateri izdelki, ki uporavljajo LON:

LON-Digital input 4port (contact. voltage) (01000-LF-DI4) LF-DI4 LON is suitable for detecting potential-free switch states, for example electrical limit switches on vent valves or auxiliary contacts of power con- tactors. The input terminals 1 to 4 are wired with the C2 terminals to potential-free switches or contacts. TP/FT-10 free topology twisted pair (ISO/IEC 14908-2). Supply voltage 20 V to 28 V AC/DC,	183.26 €
LON-Digital input 10port (contact, voltage) (1108511319) The LF-DI10 LON is suitable for detecting potential-free switch states, for example electrical limit switches on vent valves or auxiliary contacts of power contractors, push buttons, window contacts, dew point sensors, occupancy sensors. TP/FT-10 free topology twisted pair (ISO/IEC 14908-2), Supply voltage 20 V to 28 V AC/DC	192.67 €
LON-Digital input 4port (230V) (01000-LF-DI230) The LF-DI230 LON is suitable for detecting 230 V AC switch states, for example, switches or buttons for light control. The input terminals 1L to 4L are wired with 1N to 4N terminals to 230 V AC via switches or contacts. TP/FT-10 free topology twisted pair (ISO/IEC 14908-2), supply voltage 20 V to 28 V AC/DC	198.54€
LON-4 50 inputs (DIN EN 62053-31) (01000-LF-SI4) The LF-SI4 LON is suitable for counting 50 counter pulses. The software contains the LONMARK profile 2201-10 utility meter. This allows very good integration of the module into a LON-based energy controlling system. For each channel, the module saves up to 500 data records consisting of counter pulses and time stamps by means of a real-time clock (RTC). This makes it possible to use the LF-SI4 also as data logger. In case of a power failure, the data records remain saved. TP/FT-10 free topology twisted pair (ISO/IEC 14908-2), Supply voltage 20 V to 28 V AC/DC	219.67€
LON-Analogue input 4port (current, voltage) (01000-LF-CI4) LF-CI4 LON analogue sensor module with 4 current or 4 voltage inputs for active 0-20 mA and 0-10V sensors (temperature, electrical vent, valve positions, window monitoring, dewpoint monitoring). TP/FT-10 free topology twisted pair (ISO/IEC 14908-2), Supply voltage 20 V to 28 V AC/DC	238.45 €
LON-Analogue input 8port (resistance, voltage) (01000-LF-AI8) LF-AI8 LON analogue input for passive temperature sensors (Pt1000, Ni1000, NTC) and other resistances and active 0-10V sensors (temperature, electrical vent, valve positions, window monitoring, dewpoint monitoring). TP/FT-10 free topology twisted pair (ISO/IEC 14908-2), supply voltage 20 V to 28 V AC/DC	292.50 €
LON-Mixed (digital in, digital out 16A) (01000-LF-DIO4-2) LF-DIO4/2 LON sensor/actuator module with 4 digital inputs (for example for light switches and window contacts in a room) and 2 relay outputs for switching two light strips or controlling louvers. It can also be used to control 2 motorized fire dampers. TP/FT-10 free topology twisted pair (ISO/IEC 14908-2). Supply voltage 20 V to 28 V AC/DC	258.46 €
LON-Mixed (digital in, digital out 10A) (01000-LF-DIO4-2-IP65) LF-DIO4/2-IP65 LON sensor/actuator module with 4 digital inputs (for example for light switches and window contacts in a room) and 2 relay outputs for switching two light strips or controlling louvers. It can also be used to control 2 motorized fire dampers. TP/FT-10 free topology twisted pair (ISO/IEC 14908-2), Supply voltage 20 V to 28 V AC/DC	285.46 €
LON-Mixed (digital in, digital out 6A and 100 mA) (01000-DM44) LF-DM4/4 LON sensor/actuator module with 4 digital inputs (for example for light switches and window contacts in a room) and 2 relay outputs and 2 digital outputs for switching two light strips. It can also be used to control motorized fire dampers. TP/FT-10 free topology twisted pair (ISO/IEC 14908-2), Supply voltage 20 V to 28 V AC/DC With strong inductive loads, it's recommended to protect the relay	213.80 €

Slika 43: Cene za LON

Vidimo lahko, da so ti izdelki precej dragi in niso pod ceno 150-200€ za modul, to je cena za le en modul in samo za ON-OFF varianto, mi bi jih potrebovali za izdelavo enakih stropnih luči kar nekaj, zato smo izračunali, da neglede na to ali bi uporabili DALI ali LON bi nas to stalo za 42 led luči okoli dobrih 10.000,00€. V našem primeru pa smo za vse skupaj porabili manj kot 1.800,00€ za vseh 42 luči, krmilno elektroniko in še displej ter to z delom brez montaže.

V tabeli 2 je pr	rimerjava naše	tržne cene z cenami	z DALI od Schneider	electronic.
------------------	----------------	---------------------	---------------------	-------------

DALI	Cena	Št.	Cena	Naš izdelek	Cena za	Št.	Cena
	za 1kos	kos	skupaj		1kos	kos	skupaj
LED luč	80,00€	42	3.360,00€	Led luč	35,00€	42	1.470,00€
				S			
				komunikacijo			
DALI modul	105,00€	42	4.410,00€	Ne	0	0	0
ON-OFF				potrebujemo			
(Nima PWM)							
DALI gateway	301,00€	5	1.505,00€	LCD	160,00€	1	160,00€
Usmernik za	39,00€	42	1.638,00€	Usmernik	67,00€	1	67,00€
DALI luči				230Vac	-		
				/12V 12A			
DALI			Naša izdelek				
Cena skupaj brez montaže		<u>10.913,00€</u>	Cena skupaj brez montaže			<u>1.697,00€</u>	

Tabela 2

4 SKLEPI

Ugotovimo lahko, da je naš izdelek konkurenčen že obstoječim na tržišču, saj je prav tako dober, kot že obstoječi izdelki vendar bistveno cenejši zaradi komunikacije, ki smo jo uporabili pa tudi zato, ker smo vse od ohišja za led luči, krmilne elektronike ter krmilnega displeja izdelali sami in s tem privarčevali.

5 DRUŽBENA ODGOVORNOST:

V tem delu bi zapisali, da je izdelava naših led luči cenejša od drugih, kar prispeva k dostopnosti pametnih inštalacij širšemu obsegu ljudi tudi tistim, ki si dražjih opcij ne morajo privoščiti. Tako prispevamo k družbi, da omogočimo moderniziranje hiš in drugih objektov tudi tistim, ki si prej tega niso mogli privoščiti zaradi visokih cen drugih proizvajalcev. S tem izdelkom, spodbujamo Slovensko obrt oziroma družinska podjetja saj bi z sodelovanjem prodajalcev in montažerjev lahko uspešno delala manjša skupina obrtnikov oziroma lahko tudi študentov preko študentskih napotnic. Izdelek je zelo varen saj uporablja izhodno napetost 12V, ki ni nevarna ljudem tako kot je to 230Vac.

6 ZAKLJUČEK:

V zaključku bi zapisali da nam je bilo izdelovanje tega praktičnega izdelka zelo zanimivo in smo se naučili marsikaj novega. Izpolnili smo tudi vse cilje, ki smo si jih zadali ti pa so bili izdelati pametne LED luči in krmilni displej na dotik. Oboje smo uspešno izdelali, zmontirali in preizkusili delovanje. Uspelo pa nam je narediti tudi cenejši izdelek od konkurence kar pomeni, da smo prispevali k širjenju moderniziranja bivalnih objektov kot smo že prej omenili. Naš glavni cilj pa je bil najti svoj način upravljanja LED luči s pomočjo krmilnega displeja na pregleden in enostaven način po lastnem protokolu, ki nam omogoča nadgradnjo našega sistema kot je na primer odziv luči na različne senzorje kot so na primer senzorji gibanja in svetlobe. Senzorjev gibanja v našem primeru nismo dodali saj je prostor v našem primeru premajhen in bi bilo kaj takega neuporabno, uporabili pa bi jih lahko na primer na hodnikih. V prihodnje pa bomo definitivno dodali senzorje svetlobe, ki nam bodo prilagajali svetilnost luči glede na zunanjo svetlobo. Za takšno postavitev luči smo se odločili ker je bilo to najenostavnejše za polaganje kablov sicer bi pri tem delu naloge porabili še več časa. Na koncu pri testiranju smo ugotovili, da vse deluje tako kot smo si zamislili in smo z izdelkom zadovoljni. Naš izdelek je konkurenčno zelo zanimiv zato ga bomo v prihodnje še dodelali. Ideja za prihodnost je priklop BLE modula na LCD ter upravljanje luči preko mobilnega telefona, kar je danes moderno.

7 VIRI IN LITERATURA:

7.1 Spletni viri:

- Displej 4d-systemy 2023.(2.11.2023). Dostopno na: https://4dsystems.com.au/
- Displej 4d-systemy 2023.(2.11.2023). Dostopno na: https://4dsystems.com.au/products/featured-products/gen4-ulcd-43dt
- Displej 4d-systemy 2023.(2.11.2023). Dostopno na: https://4dsystems.com.au/products/featured-products/gen4-ulcd-43dt
- ST-microcontrollers 2023.(2.12.2023). Dostopno na: https://www.st.com/en/development-tools/stm32cubemx.html
- ST-microcontrollers 2023.(2.12.2023). Dostopno na: https://www.st.com/en/microcontrollers-microprocessors/stm32g030j6.html
- Program za programiranje v C 2023.(9.12.2023). Dostopno na: https://www.iar.com/products/architectures/arm/iar-embedded-workbench-for-arm/
- Digitalni izvod knjige za učenje programiranja v C 2023.(9.12.2023). Dostopno na: <u>https://cplusplus.com/doc/</u>
- DALI quick start guide 2023.(2.11.2023). Dostopno na: <u>https://www.dali-alliance.org/data/downloadables/6/4/dali-quick-start-guide_public-v1_april-2018.pdf</u>
- LON protocol 2023.(2.11.2023). Dostopno na: <u>https://mosinv.ru/Documentation/FREE/EVS/9MA10266_User%20Guide%20EVS%2</u> <u>0LON%20Application%20Notes%20-%20EN_22.09.16.pdf</u>

7.2 Knjižni viri:

- Burkhard Kainka, Basic electronics for beginners, 2020, London, Elektor International Media b.v.
- Tam Hanna, Microcontroller Basic with PIC, 2020, London, Elektor International Media b.v.
- Dogan Ibrahim, STM32 nucleo boards, 2014, London, Elektor International Media b.v.
- Herbert Schildt, C fourth edition, 2018, Berkeley Californija ZDA, Osborne