

**SVEUČILIŠTE U SPLITU  
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I  
BRODOGRADNJE**

**ZAVRŠNI RAD**

**MAPIRANJE PROSTORA POMOĆU LIDAR  
SENZORA**

**Josip Krišto**

Split, srpanj 2016.



SVEUČILIŠTE U SPLITU  
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I  
BRODOGRADNJE



Prediplomski studij: **Elektrotehnika i informacijska tehnologija**

Smjer/Usmjerenje: **Automatika i sustavi**

Oznaka programa: 111

Akadska godina: 2015./2016.

Ime i prezime: **Josip Krišto**

Broj indeksa: 173-2013

## **ZADATAK ZAVRŠNOG RADA**

Naslov: **Mapiranje prostora pomoću LIDAR senzora**

Zadatak: U završnom radu potrebno je izraditi sustav za mapiranje prostora koristeći tehnologiju koja se temelji na osvjetljavanju prepreka laserom. Prilagoditi postojeći ili izraditi softver za vizualizaciju rezultata mapiranja. Pri realizaciji koristiti LIDAR-Lite senzor. Rezultate rada sustava provjeriti u realnim uvjetima te ih komentirati.

Prijava rada: 29.02.2016.

Rok za predaju rada: 23.09.2016.

Rad predan:

Mentor:

Prof. dr. sc. Vladan Papić

# SADRŽAJ

<b>1. UVOD .....</b>	<b>1</b>
<b>2. LIDAR LITE v2 „BLUE LABEL“ .....</b>	<b>2</b>
<b>2.1. Općenito o LIDAR-Lite v2 .....</b>	<b>2</b>
2.1.1. Karakteristike LIDAR-Lite v2 .....	3
2.1.2. LIDAR-Lite v2 definicija sučelja signala i napajanja .....	4
<b>2.2. Tehnologija i hardverski sustav .....</b>	<b>6</b>
2.2.1. Tehnologija .....	6
2.2.2. Hardverski sustav .....	6
2.2.3. Jezgra procesiranja signala (SPC) .....	7
2.2.4. Optički odašiljač i prijamnik .....	8
2.2.5. Uslovi napajanja .....	9
<b>2.3. LIDAR-Lite v2 u radu .....</b>	<b>9</b>
2.3.1. Pin za upravljanje načinom rada .....	10
2.3.2. Postavke akvizicije .....	11
2.3.3. Korelacijski zapis .....	11
2.3.4. Uspostavljanje veze za I2C i PWM .....	14
<b>3. HARDVER .....</b>	<b>16</b>
<b>3.1. Arduino .....</b>	<b>16</b>
3.1.1. Općenito o arduinu .....	16
3.1.2. Arduino Uno .....	17
<b>3.2. Servo motori .....</b>	<b>19</b>
<b>4. SOFTVER .....</b>	<b>21</b>
<b>4.1. Arduino razvojna okolina (IDE) .....</b>	<b>21</b>
<b>4.2. Processing .....</b>	<b>23</b>

<b>5. MAPIRANJE PROSTORA POMOĆU LIDAR LITE v2 .....</b>	<b>25</b>
<b>6. ZAKLJUČAK .....</b>	<b>39</b>
<b>LITERATURA .....</b>	<b>40</b>
<b>POPIS OZNAKA I KRATICA .....</b>	<b>41</b>
<b>SAŽETAK .....</b>	<b>42</b>

## 1. UVOD

LIDAR (akronim od engl. Light Detection And Ranging) je optički mjerni instrument koji mjeri udaljenost osvijetljavajući metu laserskim zrakama. Ima široku primjenu u mjernim instrumentima za mjerenje podnice i unutarnje strukture oblaka, otkrivanje turbulencije u bezoblačnom zraku, u istraživanju mikrofizike oblaka te onečišćenja zraka. LIDAR se popularno koristi kao tehnologija za stvaranje mapa velike rezolucije, sa primjenom u geodeziji, geomatici, arheologiji, geografiji, geologiji, geomorfologiji, seizmologiji, šumarstvu, atmosferskoj fizici i laserske altimetrije [1]. Drugi naziv za LIDAR je optički radar (engl. Light Radar).

LIDAR koristi ultraljubičasto, vidljivo ili skoro infracrveno svjetlo za stvaranje slike objekta. Može ciljati širok opseg materijala, uključujući nemetalne objekte, stijene, kišu, kemijske spojeve, aerosole, oblake te čak i same molekule. Uska laserska zraka može mapirati fizičke karakteristike sa vrlo visokom rezolucijom; npr. letjelica može mapirati teren sa rezolucijom od 30 cm ili boljom [2].

U ovom završnom radu cilj je upotrebom LIDAR-Lite v2 te ostalih potrebnih komponenti za njegovo funkcioniranje, pomoću izmjerenih udaljenosti predmeta te svih ostalih točki u željenom prostoru stvoriti i vizualizirati mapu samog prostora. U ovom slučaju koristit će se 2 servo SG90 motora koji će se nalaziti u plastičnom montiranom kućištu (pan tilt), gdje će jedan servo motor služiti za okretanje LIDAR-a u smjeru x-osi dok će drugi služiti za okretanje LIDAR-a u smjeru y-osi. Također na isto kućište biti će pričvršćen LIDAR-Lite v2. Dalje u ovom slučaju koristi će se arduino uno r3 preko kojega ćemo upravljati motorima i samim LIDAR-om. Arduino će služiti za napajanje servo motora i LIDAR-a te će se pomoću I2C adresiranja komunicirati s LIDAR-om te primati i obrađivati podatke. Jedan od programa koji će se koristiti za primanje podataka od arduina te same vizualizacije tih podataka je proccesing. Razradit će se detaljno tehnologije koje su korištene za rad, LIDAR-Lite v2, arduino uno r3 kao i arduinova razvojna okolina te proccesing. Prikazat će se i programski kod korišten za upravljanje LIDAR-om kao i kod za realizaciju vizualizacije te će se detaljno objasniti.

## 2. LIDAR-LITE v2 „BLUE LABEL“

### 2.1 Općenito o LIDAR-Lite v2

LIDAR-Lite v2 prikazan na slici 2.1.1 je kompaktni optički senzor visokih performansi za mjerenje udaljenosti. LIDAR-Lite „BLUE LABEL“ je idealan za korištenje u dronovima, robotima ili u bezposadnim vozilima u situacijama kada je potreban pouzdan i moćan senzor blizine (udaljenosti) a da ne zauzima previše prostora. Sve što je potrebno za komuniciranje sa ovim senzorom je standardno I<sup>2</sup>C ili PWM sučelje. Sa svime spojenim LIDAR-Lite v2 sa svojim dometom od 40 metara bit će spreman za rad [2].



*Slika 2.1. LIDAR-Lite v2*

### 2.1.1 Karakteristike LIDAR-Lite v2

*Tablica 2.1. Osnovni podaci LIDAR-Lite v2 [3]*

Općenito	Tehnički podaci
Napajanje	4,75-5,5 V DC nazivni, maksimalno 6 V DC
Težina	PCB 4,5 grama, modul 22 grama sa optikom i kućištem
Veličina	PCB 44,5 X 16,5mm (1,75" sa 0,65")
Kućište	20 X 48 X 40mm (0,85" X 1,9" X 1,6")
Potrošnja struje	<2 mA @ 1 Hz (prestanak rada između mjerenja), <100 mA (uzastopni rad)
Max. Temperatura rada	70° C
Vanjski okidač	3,3 V logično, visoka-niska rubno okidano
PWM domet izlaza	PWM (Pulse Width Modulation) signal proporcionalan dometu, 1 msec/metar, 10 µsec veličina koraka
I2C sučelje uređaja	100Kb – fiksirano, 0xC4 „slave“ adresa. Pristup i upravljanje unutarnjeg registra.
Podržane I2C naredbe	Mjerenje jedne udaljenosti, brzina, snaga signala
Način upravljanja	Zauzet status korištenjem I2C, ulaz vanjskog okidača / PWM izlazi
Max. Domet pod tipičnim uvjetima	~40 m
Preciznost	+/- 2,5 cm, ili +/- ~1"
Zadana brzina pulseva	~50 Hz

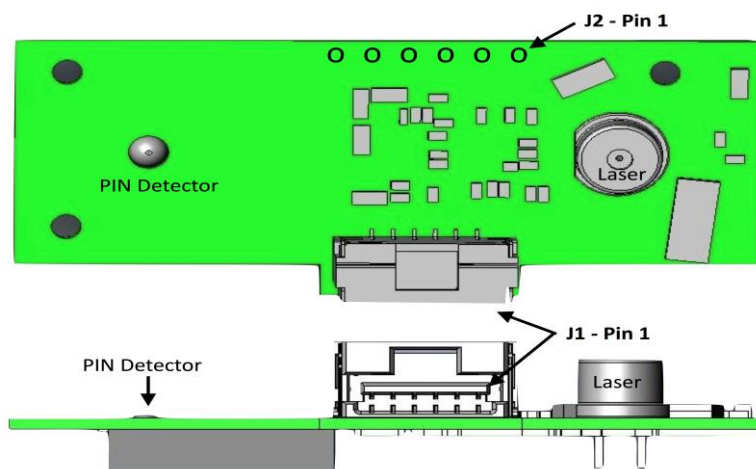
## Klasa 1 laserski proizvod

Ovaj laserski proizvod je određen kao klasa 1 tijekom svih postupaka rada. To znači da je laser siguran za gledati sa zdravim okom. Kakogod, vrlo je preporučljivo izbjegavati gledanje u zrake i isključiti modul kada nije u upotrebi.

Tablica 2.2. Karakteristike lasera [3]

Parametri	Vrijednosti lasera
Valna duljina	905 nm (nazivna)
Ukupna snaga lasera-vrh	1,3 W
Način rada	Pulsovi (max. 256 pulsova)
Širina pulsa	0,5 $\mu$ Sec (50% radnog ciklusa)
Frekvencija ponavljanja pulsa	10-20 KHz nazivna
Energija po pulsu	<280 nj
Promjer zrake kod otvora lasera	12 mm x 2 mm
Divergencija	4 mRadijan x 2 mRadijan (približno)

### 2.1.2 LIDAR-Lite Definicije sučelja signala i napajanja



Slika 2.2. Smještaj sučelja i dijelova LIDAR-a na pločici [3]



*Tablica 2.3. Opis pinova primarnog sučelja J1 [3]*

Pin	Opis
PIN1	Ulaz napajanja – 4,75-5,5 V DC nazivno, maksimalno 6 V DC. Maksimalno crpljenje struje iz ovog ulaza (koje se događa tokom perioda uzimanja podataka) je tipično <100 mA u trajanju od 4 do 20 ms ovisno o snazi primljenog signala.
PIN2	Omogućeno napajanje – aktivno visoka razina, omogućava rad 3,3 V regulatora mikro-kontrolera. Niska razina gasi pločicu, vuče <40 mA. (Unutarnji 100 K pull-up)
PIN3	Način rada – omogućava okidanje (visoki-niski rub) PWM izlaza (visoka)
PIN4	I2C takt (SCL)
PIN5	I2C podaci (SDA)
PIN6	Signal/napajanje uzemljenje

*Tablica 2.4. Opis pinova sekundarnog sučelja signal/napajanje (tvorničke postavke) [3]*

Pin	Opis
PIN1	Laser baj-pas 5-20 V max. (nazivno spojen na pin 2 preko zavojnice L8 – uklonjen zbog vanjskog napajanja)
PIN2	Ulazno napajanje – 4,75-5,5 V DC nazivno, maksimalno 6 V DC
PIN3	Omogućeno napajanje – aktivna visoka razina
PIN4	Ulaz vanjskog referentnog takta (tvornička postavka)
PIN5	Signal/napajanje uzemljenje
PIN6	Radna točka detektora-do 25 V vanjskog radnog napona za PIN, ulaz vanjskog radnog 200 V za APD

## **2.2 Tehnologija i Hardverski sustav**

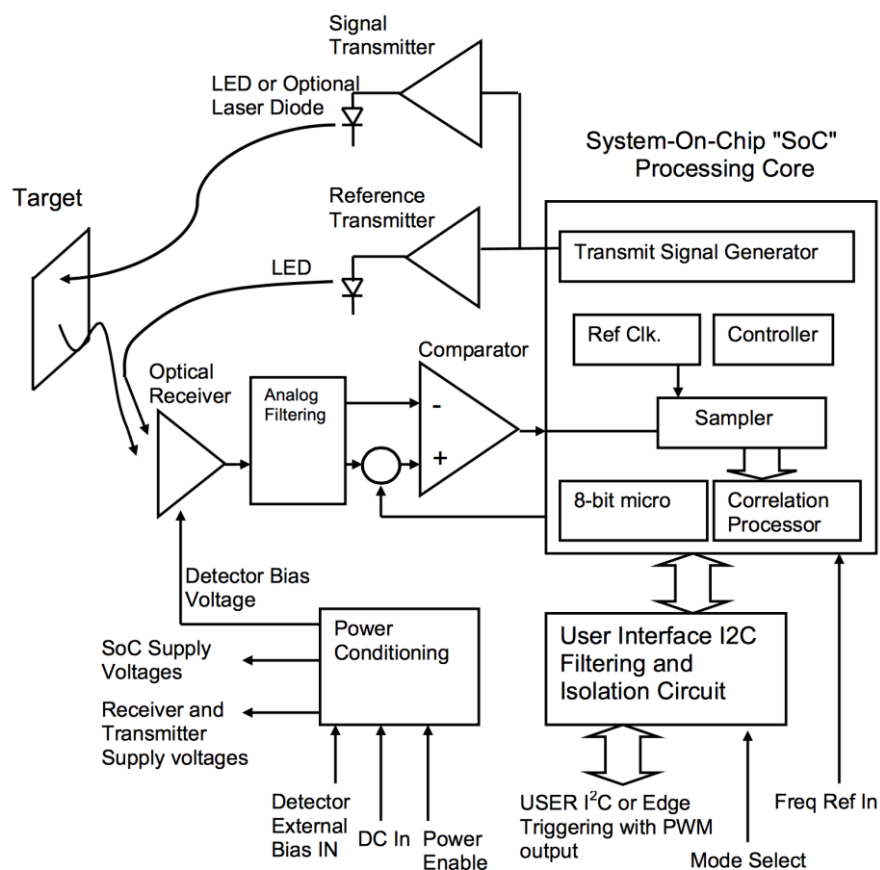
### **2.2.1 Tehnologija**

Pulselight-ova „vrijeme-leta“ tehnologija za mjerenje udaljenosti je bazirana na preciznim mjerenjima vremenskog kašnjenja odašiljanog optičkog signala i njegovog prijema. Ova visoko precizna tehnika mjerenja digitalizacijom i izračunavanjem prosječne vrijednosti dva signala; referentnog signala poslanog od odašiljača i primljenog signala koji se odbio od mete, omogućava rezoluciju izmjerene udaljenosti od 1 cm. Vrijeme kašnjenja između ova 2 uskladištena signala je procijenjeno načinom procesiranja signala zvanim korelacija. Ovaj korelacijski algoritam precizno računa vrijeme kašnjenja, koje se preračunava u udaljenost bazirano na poznatoj brzini svjetla [3].

### **2.2.2 Hardverski sustav**

Senzor omogućava mjerenja brzine i udaljenosti u vrlo malom faktoru oblika. Ova mala veličina je rezultat system-on-chip (SoC) tehnologiji procesiranja signala, koja osim što je mala, smanjuje kompleksnost i potrošnju potpornog strujnog kruga. Sustav se sastoji od 3 ključne funkcionalnosti:

- 1) Jezgra procesiranja signala (SPC) System-on-chip rješenje enkapsulira sve potrebne funkcije u podržavanju arhitekture sustava pronalaženja dometa.
- 2) Optički odašiljač i prijamnik vezan za SPC emitira i prima uzorak optičkog signala generiranog od SPC-a
- 3) Upravljanje napajanjem i filtriranje te poliranje I2C signala



Slika 2.3. Blok dijagram LIDAR-Lite v2 [3]

### 2.2.3 Jezgra procesiranja signala (SPC)

SPC sadrži 4 glavna podsustava:

- 1) 8-bitni mikrokontroler omogućava upravljanje sustavom i komunikaciju.
- 2) 500 MHz takt uzorkovanja i povezani uređaj za uzorkovanje hvataju logično stanje vanjskog komparatora i pretvara podatke u manju brzinu 125 MHz 4-bitnu riječ koja se pošalje procesoru za korelaciju.
- 3) Korelacijski procesor pohranjuje dolazeći signal i izvodi operaciju korelacije protiv referentnim uskladištenim signalom sa prijemom optičkog praska i pohranjuje rezultat u korelacijsku memoriju sa podacima svake 2 ns.
- 4) Generator odašiljanog signala stvara kodirani valni oblik signala sa ukupnim trajanjem od 500 ns koji se sastoji od varirajući uzoraka nula i jedinica. Ovi odlazeći signalni pulsovi događaju se pri 20 KHz stopom ponavljanja [3].

## 2.2.4 Optički odašiljač i prijamnik

Odašiljač stvara optičke izboje pulsova koristeći uzorke signala koje generira SPC. Kada je referentni optički signal poželjan, odvojeni referentni odašiljač je omogućen i upravljan sa uzorkom signala koristeći referentni izvor prema optičkom prijamniku. Referentni odašiljač je dizajniran tako da odgovara kašnjenju i obliku signala generiranom od odašiljača signala veće snage. Odašiljač signala može pogoniti raznoliko optičkih izvora u rasponu od jako brzih LED-ica, snažnih VCSEL lasera ili mnogo snažnijih pulsirajućih laserskih dioda. Za LIDAR-Lite modul, driver odašiljanog signala pogoni T1-3/4 plastično pakiranu lasersku diodu sa maksimumom od 3 A, 50 %-tnim prosječnim radnim ciklusom modulacije u trajanju izboja od 500 ns. Driver ima mogućnost da pogoni izvore od 6 A koristeći vanjsko DC napajanje [3].

*Tablica 2.5. Karakteristike odašiljača [3]*

Parametar	Karakteristike odašiljača
Propusnost	50 MHz, sa/bez modulacije, proizvoljan uzorak
Vrijeme/brzina izboja	500 ns/20 KHz
Tipologija	Visoka strana strujni izvor (programibilan), niska strana diferencijalno upravljanje strujom
Referentni kanal	1 A vrh (nazivna vrijednost)
Signalni kanal	3 A vrh (nazivna vrijednost)
Prijenos snage	Upravljanje 16 koraka svaki kanal
Porast/pad	4 ns

Prijamnik uključuje predpojačalo niskog šuma koje je spojeno na PIN fotodiode ili po želji na lavinsku fotodiodu (APD). Kada se koristi APD visokih performansi potreban je vanjsko regulirani visoko naponski radni napon. APD se koristi radi povećanja osjetljivosti sustava omogućavajući time povećani domet rada ili smanjenje vremena mjerenja. Prije postizanja visoko-brzog digitalnog komparatora specijalizirano analogno filtriranje oblikuje povratni signal podrijetlom iz izlaza predpojačala [3].

Tablica 2.6. Karakteristike prijamnika [3]

Parametar	Karakteristike prijamnika
Propusnost	
Detektor	PIN dioda, 500 $\mu\text{m}$ x 500 $\mu\text{m}$ , 1,5 pF, 1,8 mm promjer leća
Virtualna veličina detektora	1 mm-ugrubo 2X povećanje leća paketa
Radni napon detektora	8 V DC nazivni, vanjski
Razina šuma predpojačala	1 pA/Hz - 2
Pojačanje impedancije	40 Kohm
Ekvivalentna snaga šuma	12 nW rms

### 2.2.5 Uslovi napajanja

Potrebni su višestruki referentni naponi za raznovrsne funkcije na LIDAR-Lite ploči. Standardni PIN detektor zahtjeva DC radni napon od 8V generiranog od unutarnjeg multiplikatora napona. Korištenje APD detektora zahtjeva temperaturno ovisni radni napon od 100 V do 240 V ovisno o izabranom detektoru. Ovaj potrebnii napon je promjenjiv bazirano na koeficijentu temperaturne kompenzacije i primijenjen kroz ulazni pin vanjskog detektora. Potrebna je tvornička modifikacija da bi se dozvolila vanjska primjena napona detektora iznad 30 V DC. Sklopovlje prijamnika koriste napajanje od 3,7 V koje je omogućeno od SPC. Sklopovlje odašiljača tipično koristi napon napajanja od 5V. Pin napajanja je povezan sa ulazom od 5 V pomoću izolirane zavojnice. Pin za omogućavanje dozvoljava unutarnjem regulatoru od 3,3 V da bude onesposobljen radi jako male potrošnje pri uvjetima isključivanja [3].

### 2.3 LIDAR-Lite v2 u radu

Rad LIDAR-lite-a može biti podijeljen u 2 faze: inicijalizaciju i izazvanu akviziciju pokrenutu od korisnika.

Tijekom inicijalizacije mikrokontroler prolazi kroz slijed osobnih testova praćenih inicijalizacijom unutarnjih upravljačkih registara sa početnim vrijednostima. Unutarnji upravljački registri mogu biti prilagođeni od strane korisnika pomoću I2C sučelja nakon inicijalizacije.

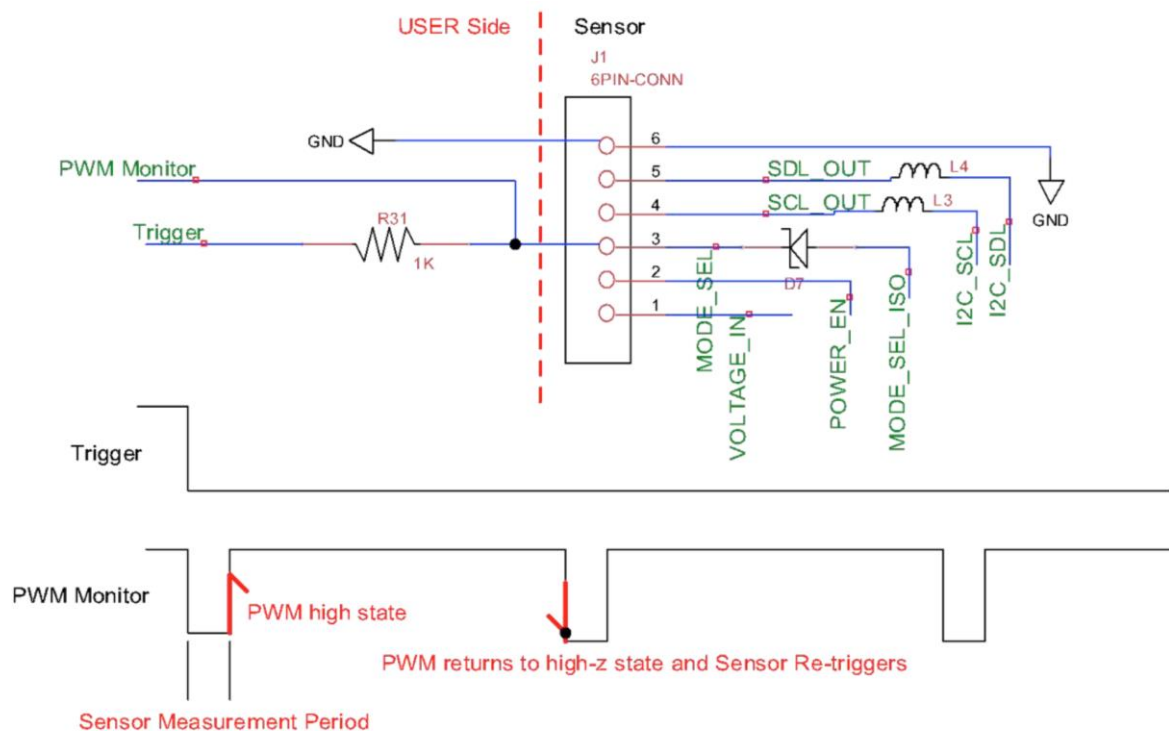
Nakon što su unutarnji upravljački registri inicijalizirani procesor ide u stanje „spavanja“ smanjujući ukupnu potrošnju ispod 10 mA. Korisničkom naredbom pomoću vanjskog okidanja ili I2C naredbe, budi procesor omogućujući sljedeću operaciju [3].

Unos naredbe preko I2C sučelja može pokrenuti akviziciju ili operaciju za motrenje ili modificiranje parametara sustava. U slučaju zahtjeva akvizicije, sustav se mora prvo upaliti i inicijalizirati vanjske funkcije kao što su SPC i sklopovlje za odašiljanje/primanje. Akvizicija počinje sa prijenosom referentnog praska praćenog signalnim praskom. Signalni praskovi događaju se u intervalu od 50-100  $\mu$ s ovisno o duljini izabranog korelacijskog zapisa. Signalni praskovi se ponavljaju dok maksimalni broj akvizicija ne bude postignut, što je definirano kao početno ili u korisničkim postavkama, ili dok dovoljan broj akvizicija ne bude postignut da bi se dobila maksimalna razina snage signala. Nakon toga rezultati korelacije se procesiraju da bi se izračunalo efektivno vrijeme kašnjenja referentnog i dobivenog signala sa korelacijskim zapisom. Ukupno vrijeme akvizicije za referencu i akvizicije signala je tipično između 5 i 20 ms ovisno o željenom broju pulseva i duljini korelacijskog zapisa. Vrijeme akvizicije plus potrebna 1 msec za skidanje parametara mjerenja uspostavljaju približno maksimalnu brzinu mjerenja od 100 Hz [3].

### 2.3.1 Pin za upravljanje načinom rada

Dvosmjerni pin za upravljanje i stanje pruža način za pokretanje akvizicije i povratka izmjerene udaljenosti preko pulsno širinske modulacije [PWM] bez potrebe za korištenjem I2C sučelja.

Driver za pin u procesoru ima unutarnji „pull-up“ izvor struje od oko 50  $\mu$ A sa izlazom drivera povezanim na korisnički pin pomoću zaštitne diode dopuštajući jedino napajanje strujom u pin. Sporo-odlazeći prijenos na pinu za kontrolu moda će pokrenuti jedno mjerenje, i pin će aktivno vući visoko sa širinom pulsa proporcionalnoj udaljenosti. Otpornik za terminaciju od 1 K do 10 K oma će solidno vući pin u nisko za pokretanje stanja akvizicije omogućavajući pin-u da i dalje bude vučen visoko tijekom PWM izlaznog pulsa. Širina pulsa prati vezu od 10  $\mu$ sec/cm sa izmjerenom udaljenosti ili 1 msec po metru. Jednostavna metoda okidanja koristeći standardno mikrokontrolersko sučelje koristi otpornik od 1 K oma u seriji sa izlaznim pin-om za povlačenje pin-a za kontrolu moda nisko pokretajući mjerenje sa drugim port pin-om korištenim za nadgledanje niskim-visokim izlazom širine pulsa. Ako se pin drži nisko, proces akvizicije će se ponavljati beskonačno puta generirajući izlaz promjenjive frekvencije proporcionalan udaljenosti [3].



Slika 2.4. Prikaz upravljanja *MODE* pinom [3]

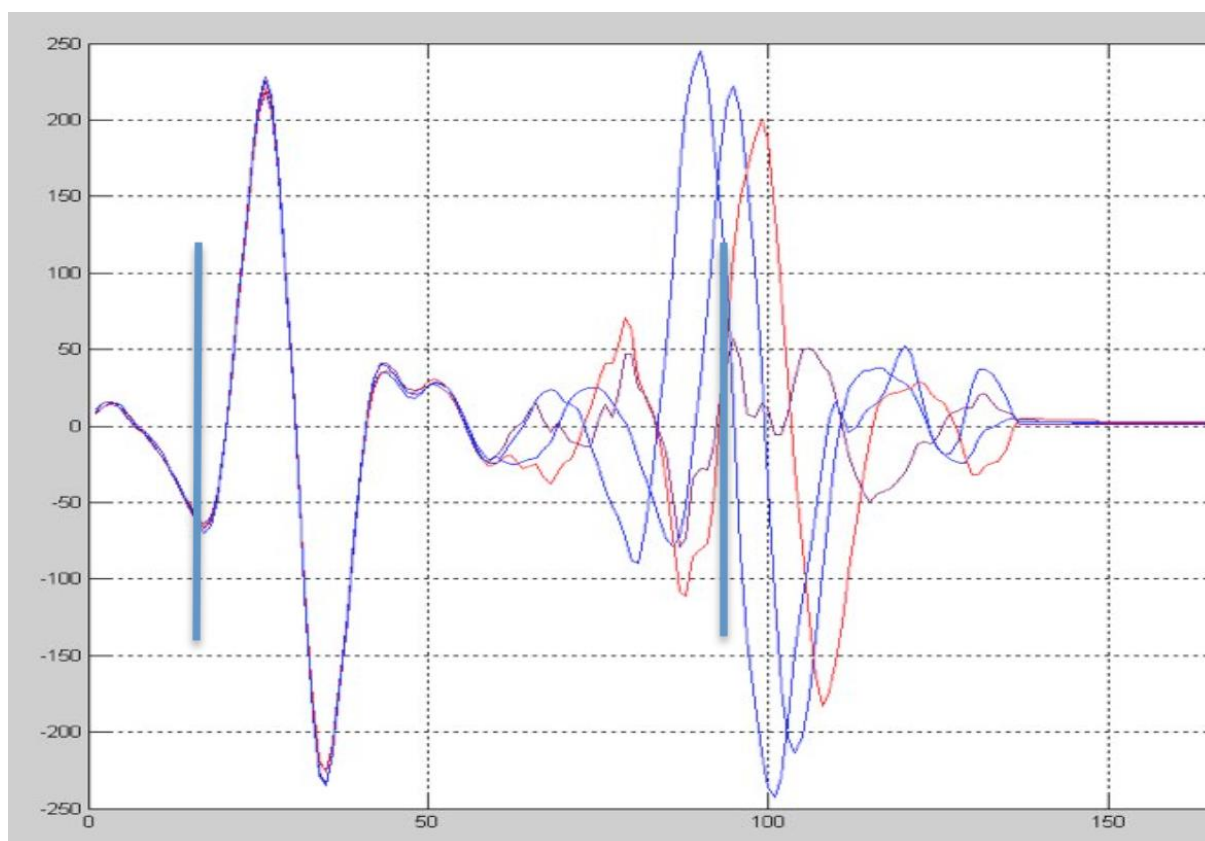
### 2.3.2 Postavke akvizicije

Parametri akvizicije signala mogu se lako promijeniti u zamjenu za parametre performansi sustava. Ako se zahtjeva velika brzina mjerenja, tada se maksimalno vrijeme integracije signala može smanjiti da bi se smanjila vremena mjerenja u trošku od malo smanjene osjetljivosti i maksimalnog dometa. Optička snaga odašiljanja može se povećati pomoću postavki utrpanim u registru snage lasera. Velika snaga pulsa može trebati da bude kompenziran sa povećanim razmakom između praskova pulsa da bi se zadržao prihvatljiv radni ciklus lasera baziran na zahtjevima toplotnog prigušivanja. Ako je duljina korelacijskog zapisa povećana da bi se omogućila mjerenja većeg dometa, tada će povećano vrijeme procesiranja smanjiti brzinu mjerenja [3].

### 2.3.3 Korelacijski zapis

Mjerenja udaljenosti su bazirana na pohrani i procesiranju korelacijskih zapisa reference i signala.

Slika ispod pokazuje korelacijski zapis senzora bez optike za kratke udaljenosti od 0, 4 i 8 stopa. Referentni zapis kreće od 0-63 a signalni zapis od 64 do 130. Svaka točka uzorka predstavlja 2 nsec ili približno 1 stopu.

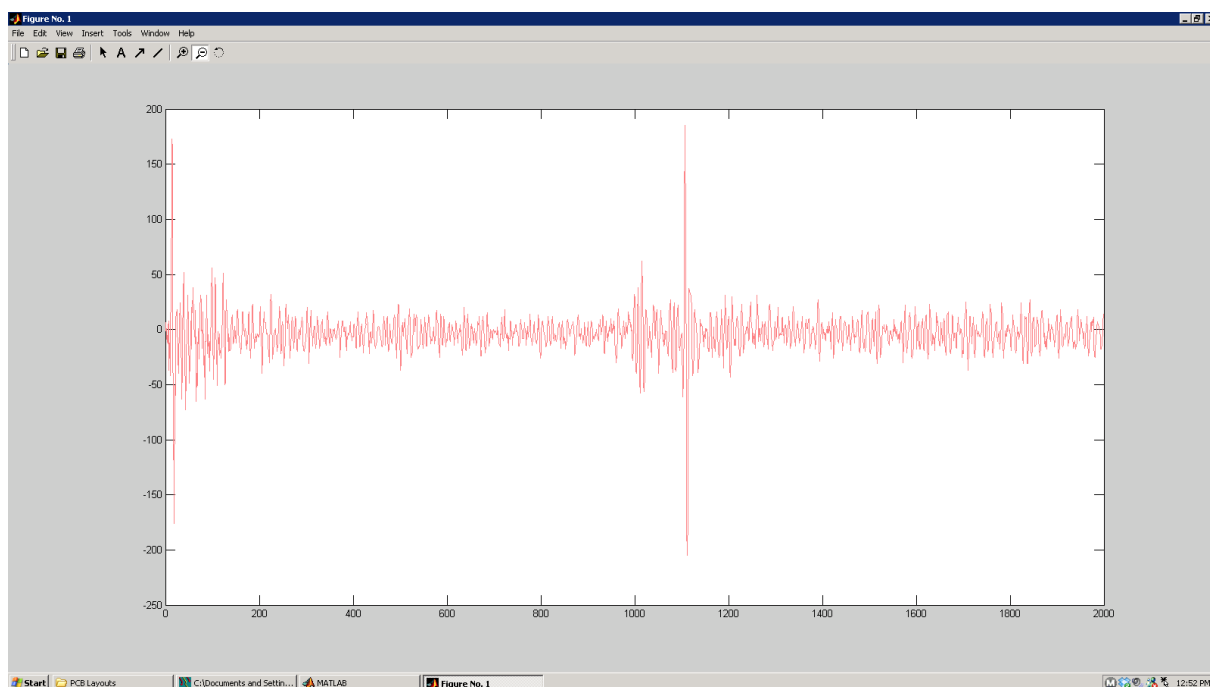


*Slika 2.5. Korelacijski zapis bez senzora za kratke udaljenosti [3]*

Korelacijski val ima bipolarni oblik vala, polazeći od pozitivnog dijela do približno simetričnog negativno-kretajućeg pulsa. Točka gdje signal prolazi kroz nulu predstavlja efektivno kašnjenje referentnih i povratnih signala. Procesiranje sa SPC određuje interpoliranu točku prelaska sa rezolucijom 1cm zajedno sa vrijednošću vrha signala [3].

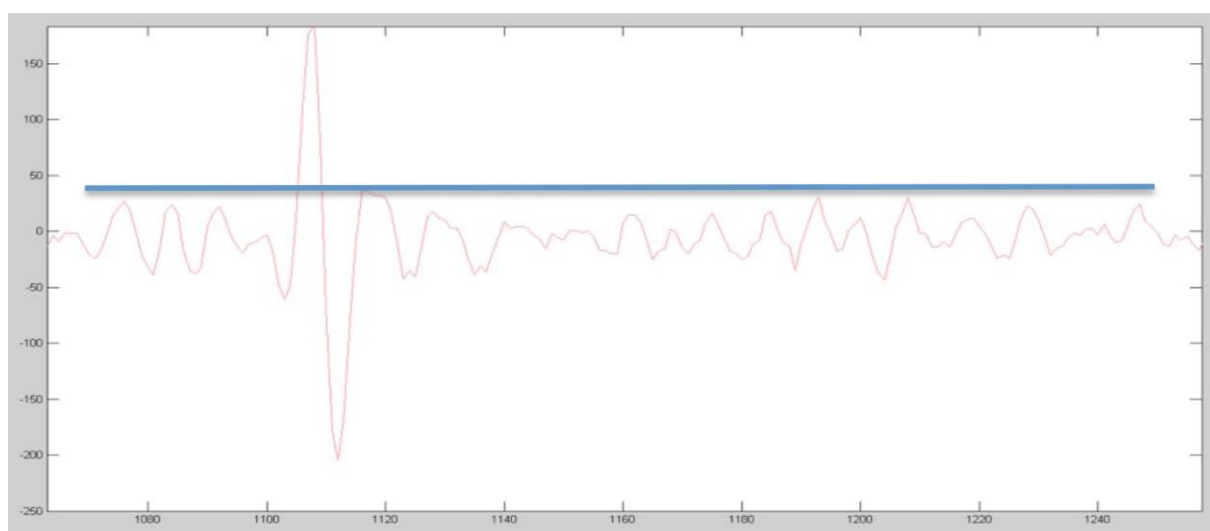
Slika ispod prikazuje primjer korelacijskog zapisa za sustave dugog dometa koristeći lavinsku fotodiodu ili APD i laser sa čipom za procesiranje sa polu-rezolucijom 2-stope/korelacijska koraka i 2 tisuće elemenata zapisa signala. Meta je 660 metara daleko i kreira isti bipolarni korelacijski valni oblik kao i kod sustava kratkog dometa, ali u praksi korelacijski val mora biti istisnut iz pozadinskog šuma prisutnog u korelacijskom zapisu.





*Slika 2.6. Korelacijski zapis sustava velikog dometa [3]*

Korelacijski val je prikazan u više detalja ispod. Da bi se izdvojio korelacijski puls od pozadinskog šuma, specijalizirani filter za procesiranje prati ovojnicu šuma bez značajnog utjecaja signalnih korelacija prisutnih u zapisu [3]. Ova referenca šuma je umanjena za 1,25 da bi se pružila detekcija praga korelacije.



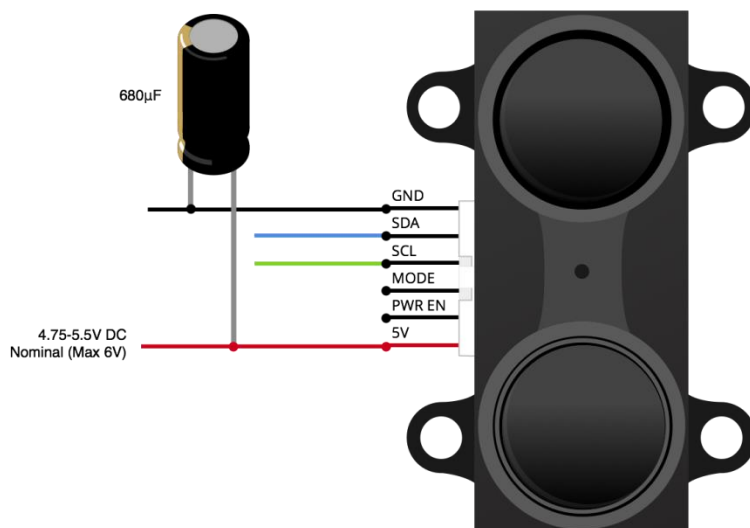
*Slika 2.7. Korelacijski zapis uvećan u više detalja [3]*

Ako je više od jednog signala detektirano sa korelacijskim zapisom, povratak sa idućim signalom najveće snage je pohranjen i slobodan za dodatno procesiranje. Zastavica sa statusnim registrom ukazuje prisustvo valjane druge refleksije kao što je od prozora ili od objekta na manjoj udaljenosti osvijetljenog zrakom. Procesiranje sekundarnog povratka je limitirano na mete slabije refleksije u prvom planu [3].

#### 2.3.4 Uspostavljanje veze za I2C i PWM

Postoje 2 osnovne konfiguracije za LIDAR-Lite:

I2C umrežavanje

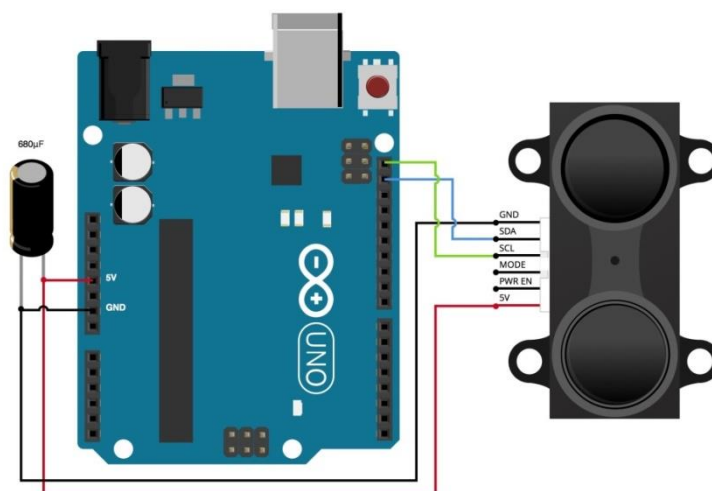


Slika 2.8. I2C umrežavanje [3]

1. Spojiti „power“ i „ground“ pinove. Senzor radi na 4,75-5,5 V DC nazivno, maksimalno 6 V DC.
2. Postaviti elektrolitski kondenzator od 680 µF između 5 V i GND
3. Spojiti I2C SCL/SDA pinove.

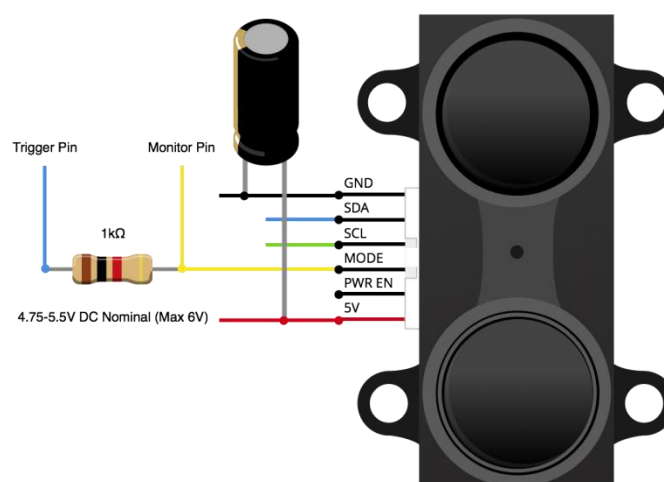
## Arduino I2C umrežavanje

Spojiti arduino kako je prikazano na slici ispod. Paziti na ispravan polaritet kondenzatora.



*Slika 2.9. Arduino I2C umrežavanje [3]*

## PWM umrežavanje

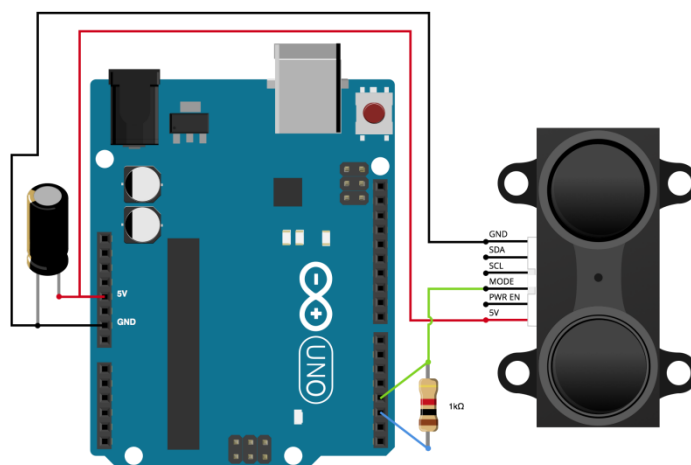


*Slika 2.10. PWM umrežavanje [3]*

1. Spojiti „power“ i „ground“ pinove. Senzor radi na 4,75-5,5 V DC nazivno, max. 6 V DC.
2. Spojiti pin MODE na otpornik od 1 kΩ i na „Monitor“ pin
3. Spojiti drugu stranu otpornika na „Trigger“ pin

## Arduino PWM umrežavanje

Spojiti arduino kao na slici ispod. Pin #2 je pin za okidanje, a pin #3 je pin za nadgledanje.



Slika 2.11. Arduino PWM umrežavanje [3]

## 3. HARDVER

### 3.1 ARDUINO

#### 3.1.1 Općenito o arduinu

Arduino je „open-source“ platforma za kreiranje prototipova bazirana na „lako za koristiti“ hardveru i softveru. Arduino ploče su u stanju čitati ulaze – svjetlo na senzoru, prst na tipki, ili poruku s twittera – i pretvoriti ga u izlaz – aktivirajući motor, paleći LED, objavljujući nečega online i dr. Može se reći ploči šta da radi slanjem seta instrukcija mikrokontroleru na ploči. Da bi to napravili koristi se arduino jezik za programiranje (baziran na Wiring-u) i arduino softver (IDE) baziran na processing-u [4].

Arduino je baziran na dizajnima mikrokontrolerske ploče, proizvedenih od nekoliko proizvođača, koristeći razne mikrokontrolere. Ovi sustavi pružaju setove digitalnih i analognih ulaznih/izlaznih (I/O) pinova koji se mogu spajati sa raznim ekspanzijskim pločama („shields“) i ostalim sklopovljem. Ploče odlikuju serijska komunikacijska sučelja, uključujući univerzalnu serijsku sabirnicu (USB) na nekim modelima, za ukrcavanje programa s osobnog računala [5].

### 3.1.2 Arduino Uno

Arduino uno je mikrokontrolerska ploča bazirana na Atmega328P. Ima 14 digitalnih ulazno/izlaznih pinova, 6 analognih ulaza, 16 MHz kvarcni kristal, USB konekciju, utikač za napajanje, ICSP zaglavlje i tipku za resetiranje. Sadrži sve potrebno za podršku mikrokontrolera; jednostavno se spoji na računalo USB kablom ili se napaja sa AC/DC adapterom ili baterijom za početak [6].

„Uno“ znači jedan na talijanskom i izabran je da označi puštanje arduino softvera (IDE) 1.0. Uno ploča i verzija 1.0 arduino softvera (IDE) bile su referentne verzije arduina, sada razvijene u novije verzije. Uno ploča je prva u seriji USB arduino ploča i referentni model za arduino platformu.

*Tablica 3.1 Tehničke karakteristike [6]*

<b>Mikrokontroler</b>	<b>Atmega328P</b>
Radni napon	5 V
Ulazni napon (preporučeni)	7-12 V
Ulazni napon (granica)	6-20 V
Digitalni I/O pinovi	14 (od kojih 6 pružaju PWM izlaz)
PWM digitalni I/O pinovi	6
Analogni ulazni pinovi	6
DC struja po I/O pinu	20 mA
DC struja za 3,3 V pin	50 mA
Flash memorija	32 KB (Atmega328P) od kojih 0,5 KB koristi „bootloader“
SRAM	2 KB (Atmega328P)
EEPROM	1 KB (Atmega328P)
Brzina takta	16 MHz
Dužina	68,6 mm
Širina	53,4 mm
Težina	25g

## Napajanje

Arduino Uno ploče mogu se napajati preko USB konekcije ili sa vanjskim napajanjem. Izvor napajanja se odabere automatski.

Vanjsko napajanje može biti AC/DC adapter ili baterija. Adapter se može spojiti uključujući 2,1 milimetarski utikač s pozitivnim centrom u utikač za napajanje na ploči. Vodovi sa baterije mogu se ubaciti u GND i Vin pinska zaglavlja POWER konektora.

Ploča može raditi na vanjskom napajanju od 6 do 20 volti. Ako je napajana sa manje od 7 V, kakogod, pin 5 V može napajati manje od 5 volti i ploča može postati nestabilna. Preporučeni raspon je od 7 do 12 volti [6].

Pinovi za napajanje su sljedeći:

- Vin – ulazni napon u arduino ploču kada se koristi vanjski izvor napajanja. Može se napajati naponom preko ovoga pina ili ako se napaja preko utikača za napajanje pristupa se preko ovoga pina.
- 5 V – ovaj pin daje reguliranih 5 V od regulatora na ploči. Ploča može biti napajana sa DC utikača za napajanje (7-12 V), USB konektora (5 V), ili sa VIN pina sa ploče (7-12 V). Napajanjem preko 5 V ili 3,3 V pinova zaobilazi regulator i može oštetiti ploču.
- 3,3 V – 3,3 volti generirano sa regulatora sa ploče. Maksimalna struja je 50 mA.
- GND – pin uzemljenja
- IOREF – Ovaj pin na arduinu pruža naponsku referencu sa kojom mikrokontroler radi. Propisno konfiguriran „shield“ može čitati napon sa IOREF pina i odabrati prikladni izvor napajanja ili omogućiti prevoditelje napona na izlazu da rade sa 5 V ili 3,3 V [6].

## Memorija

ATmega328 ima 32 KB ( 0,5 KB zauzeto „bootloader-om“). Također ima 2 KB SRAM i 1 KB EEPROM ( koji se mogu čitati i pisati sa EEPROM bibliotekom).

## Ulaz i izlaz

Svaki od 14 digitalnih pinova na Unu mogu se koristiti kao ulazi ili izlazi. Rade na 5 V. Svaki pin može pružiti ili primiti 20 mA kao preporučeni uvjeti rada i ima unutarnji pull-up otpornik od 20 do 50 k oma. Maksimum od 40 mA je vrijednost koja se ne smije premašiti ni na jednom I/O pinu da bi se izbjegla trajna šteta mikrokontrolera [6].

Neki pinovi imaju posebne funkcije:

- Serija: 0(RX) i 1 (TX). Koriste se za prijam (RX) i slanje (TX) TTL serijski podataka. Ovi pinovi su spojeni na odgovarajuće pinove ATmega8U2 USB-to-TTL serijskog čipa.
- Vanjski prekidi: 2 i 3. Ovi pinovi mogu se konfigurirati za okidanje prekida na niskoj vrijednosti, rastućem ili padajućem rubu ili promjene u vrijednosti.
- PWM: 3, 5, 6, 9, 10 i 11. Omogućuju 8-bitni PWM izlaz.
- SPI: 10(SS), 11(MOSI), 12(MISO), 13(SCK). Ovi pinovi podržavaju SPI komunikaciju koristeći SPI biblioteku.
- TWI: A4 ili SDA pin i A5 ili SCL pin. Podržavaju TWI komunikaciju koristeći Wire biblioteku [6].

Uno ima 6 analognih ulaza, označenih od A0 do A5 od kojih svaki pruža 10 bita rezolucije. Pa zadanom oni mjere od uzemljenja do 5 V iako je moguće promijeniti gornju granicu njihovog dometa koristeći AREF pin.

Ima još par drugih pinova na ploči:

- AREF – referentni napon za analogne ulaze. Koristi se sa analogReference().
- Reset – dovodeći ovu liniju nisko mikrokontroler se resetira. Tipično se koristi za dodavanje reset tipke na „shield“ [6].

### 3.2 Servo motori

Za rotaciju LIDAR-Lite oko određeni osi korištena su 2 servo SG90 9g motora.

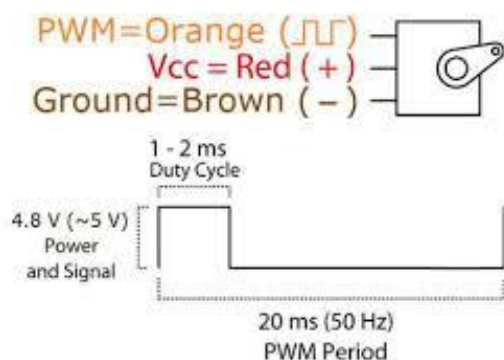


*Slika 3.1. SG90 servo motor*

Sitni i lagani sa visokom izlaznom snagom. Servo se može rotirati približno 180 stupnjeva (90 u svakom smjeru) i radi isto kao i standardni samo je manji. Može se koristiti bilo koji servo kod, hardver ili biblioteka za kontroliranje ovih servo motora. Dolazi sa 3 „roga“ i hardverom [7].

Specifikacije:

- Težina: 9g
- Dimenzije: 22,2 x 11,8 x 31 mm
- Okretna sila zaustavljanja: 1,8 kgfcm
- Brzina rada: 0,1 s/60°
- Radni napon: 4,8 V (~5 V)
- Širina mrtve zone: 10 μs
- Temperaturni raspon: 0°C - 55°C



Slika 3.2. Opis vodova servo motora i PWM signala [7]

Žuta žica je za PWM signal, crvena napajanje a smeđa uzemljenje ili masa. PWM period je 20 ms dok je radni ciklus 1 do 2 ms u iznosu od 5 V.

Pozicija „0“ (1,5 ms pulsa) je sredina, „90“ (~2 ms pulsa) je cijel put do desna, „-90“ (~1 ms pulsa) je cijel put do lijeva.



## 4. SOFTVER

### 4.1 Arduino razvojna okolina (IDE)

Arduino integrirana razvojna okolina – ili arduino softver (IDE) sadrži tekst editor (engl. text editor) za pisanje koda, područje za poruke (engl. message area), tekst konzolu (engl. text console), alatnu traku (engl. toolbar) sa tipkama za česte funkcije i niz menija. Spaja se na arduino hardver za prenošenje (engl. upload) programa i komuniciranje s njima [8].

Programi napisani koristeći arduino softver (IDE) se zovu skečevi. Ti su skečevi napisani u tekst editoru i spremljeni sa ekstenzijom podatka .ino. Editor ima svojstva za rezanje/lijepljenje i za traženje/mijenjanje teksta. Područje za poruke daje povratne informacije tijekom spremanja i izvođenja te također prikazuje greške. Konzola prikazuje izlazni tekst arduino softvera (IDE), uključujući kompletne poruke grešaka i ostale informacije. Donji desni kut prozora prikazuje konfiguriranu ploču i serijski port. Tipke alatne trake dopuštaju potvrdu i prenošenje (engl. upload) programa, otvaranje i spremanje skečeva i otvaranje serijskog monitora [8].

Arduino programi mogu se podijeliti u 3 glavna dijela:

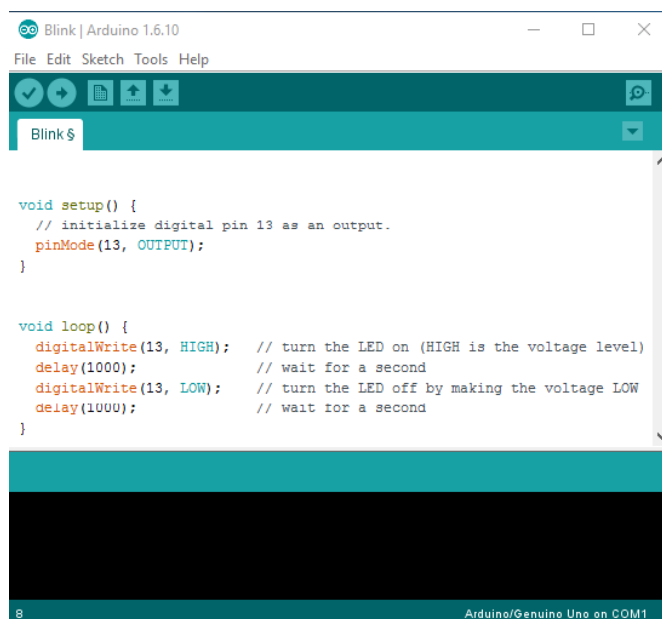
- 1) Strukture
- 2) Varijable i konstante
- 3) Funkcije

Arduino softver (IDE) je napisan u programskom jeziku Java. Arduino IDE podržava jezike C i C++ koristeći posebna pravila da bi se organizirao kod [5]. Arduino IDE sadrži softver biblioteku wiring, koja pruža mnogo čestih ulaznih i izlaznih procedura. Tipični arduino C/C++ skeč se sastoji od 2 funkcije koje su kompajlirane i povezane sa programskim dijelom *main()* u izvršno ciklički program:

- **Setup():** funkcija koja kreće jednom na početku programa i koja može inicijalizirati postavke
- **Loop():** funkcija pozivana neprestano dok se ploča ne isključi

Primjer programa:

Većina arduino ploča sadrži LED i otpornik između pina 13 i uzemljenja koji su svojstvo za mnoge testove. Tipičan program za početak programiranja u arduinu je blinkanje LED diode.



*Slika 4.1. Prikaz arduino skeča za paljenje i gašenje LED diode [8]*

## Biblioteke

Arduino razvojna okolina može biti proširena korištenjem biblioteka, kao većina platformi za programiranje. Biblioteke pružaju dodatne funkcionalnosti za korištenje u skečevima, u radu sa hardverom ili manipuliranjem podacima. Za korištenje biblioteka u skeču, odabere se iz izbornika Sketch > Import Library [9].

Dosta biblioteka dođe instalirano uz IDE, ali se naravno mogu skinuti i ostale potrebne ili napraviti svoju. Jedne od standardnih biblioteka su: Wire i Servo.

### Servo biblioteka

Ova biblioteka dopušta arduino ploči kontroliranje RC servo motora. Servo motori imaju integrirane zupčanike i osovine koje se mogu precizno kontrolirati. Standardni servo motori dopuštaju pozicioniranje osovine u raznim kutovima, obično od 0 do 180 stupnjeva. Kontinuirana rotacija motora dopušta da rotacija osovine bude podešena na različite brzine [9].

Servo biblioteka podržava do 12 motora na većini arduino ploča i 48 na arduino mega. Na pločama osim mega, korištenje biblioteka onemogućuje analogWrite() (PWM) funkcionalnost na pinovima 9 i 10, bilo na njima serva ili ne. Na mega do 12 servo motora

može biti korišteno bez uplitanja sa PWM funkcionalnostima; korištenje 12 do 23 motora će onemogućiti PWM na pinovima 11 i 12 [9].

Funkcije: attach(), write(), writeMicroseconds(), read(), attached(), detach().

Primjeri: - Knob: kontrolira osovinu servo motora zakretanjem potencijometra; - Sweep: zamahuje osovinu servo motora nazad i naprijed.

Wire biblioteka

Ova biblioteka omogućava komuniciranje sa I2C/TWI uređajima. Na arduino pločama sa r3 rasporedom, SDA (linija podataka) i SCL (linija takta) su na zaglavljima pinova blizu AREF pina. Arduino Due ima 2 I2C/TWI sučelja SDA1 i SCL1 su blizu AREF pina i dodatno je na pinovima 20 i 21 [9].

Kao referenca tablica ispod pokazuje gdje se TWI pinovi nalaze na različitim arduino pločama.

*Tablica 4.1. I2C pinovi na različitim pločama [9]*

Ploča	I2C/TWI pinovi
Uno, ethernet	A4 (SDA), A5 (SCL)
Mega2560	20 (SDA), 21 (SCL)
Leonardo	2 (SDA), 3 (SCL)
Due	20 (SDA), 21 (SCL), SDA1, SCL1

Imaju obje 7-bitne i 8-bitne verzije I2C adresiranja. 7-bitne identificiraju uređaj a 8-bitne utvrđuju jeli bilo pisano ili čitano sa njega. Wire biblioteka koristi 7-bitno adresiranje cijelo vrijeme.

Funkcije: begin(), requestFrom(), beginTransmission(), endTransmission(), write(), available(), read(), onReceive(), onRequest().

## 4.2 Processing

Općenito o processingu

Processing je programski jezik otvorenog koda (eng. open source) i integrirana razvojna okolina (IDE) napravljena za zajednicu elektroničke umjetnosti i vizualnog dizajna sa svrhom podučavanja osnova programiranja u vizualnom kontekstu i da služi kao temelj

elektroničke knjige skečeva (engl. sketchbooks) [10]. Jedan od ciljeva processinga je da omogućiti neprogramerima da započnu programirati potpomognuti vizualnim povratnim podacima. Processing je napravljen na java jeziku, ali koristi pojednostavljenu sintaksu i grafičko sučelje. Processing okolina uključuje uređivač teksta (engl. text editor), kompajler i prozor za prikazivanje. Omogućuje kreiranje softvera sa pažljivo dizajniranim setom ograničenja.

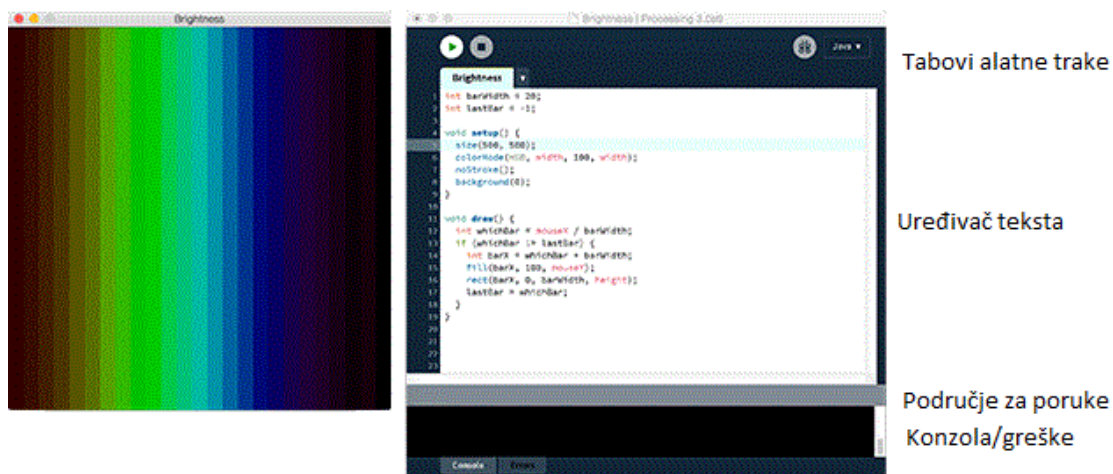
Processing programi pišu se u uređivaču teksta u processing razvojnoj okolini i započinju pritiskom na tipku pokreni (engl. run). U processingu program se zove skeč (engl. sketch). Skečevi se spremaju u knjigu skečeva (engl. sketchbook), što je mapa na računalu [11].

Skečevi mogu crtati dvodimenzionalnu i trodimenzionalnu grafiku. Zadani „renderer“ je za crtanje dvodimenzionalne grafike. P3D renderer omogućuje crtanje trodimenzionalne grafike, što uključuje kontroliranje kamere, rasvjete i materijala. Mogućnosti Processinga su proširene sa bibliotekama i alatom. Biblioteke omogućuju skečevima da rade stvari iznad jezgre processing koda. Postoji tisuće biblioteka koje su doprinos processing zajednice, koje mogu biti dodane u skečeve za omogućavanje novih stvari kao sviranje zvuka, računalne vizije i rada sa složenom 3D geometrijom. Alati proširuju processing okolinu da bi pomogli u kreiranju skečeva pružajući sučelja za zadatke kao npr. odabir boje [11].

Processing ima različite programske modove da omogućiti razvoj skečeva na različitim platformama i programima u raznim načinima. Java mod je zadani. Ostali modovi se mogu skinuti iz menija processing razvojne okoline.

#### Processing razvojna okolina (PDE)

PDE se sastoji od jednostavnog uređivača teksta za pisanje koda, područja za poruke, tekst konzole, tabova za upravljanje podacima, alatne trake sa tipkama za česte akcije i niz menija [11]. Meni opcija se mijenja ovisno o modu.



Slika 4.2. Izgled processing razvojne okoline te skeča za prikaz spektra boja [11]

Skečevi su pisani u uređivaču teksta. Ima svojstvo za rezanje/lijepljenje i traženje/mijenjanje teksta. Područje za poruke daje povratne informacije tijekom spremanja i pokretanja i prikazuje greške. Konzola prikazuje izlaz teksta processing skeča uključujući sve greške i izlaz teksta skečeva sa `print()` i `println()` funkcijama. Glavne tipke na alatnoj traci mogu pokrenuti i zaustaviti program a to su: pokreni i zaustavi tipke [11].

## 5. MAPIRANJE PROSTORA POMOĆU LIDAR-LITE v2

Tema ovoga završnog rada je bila vizualizacija izmjerenih udaljenosti LIDAR-Lite-om u svrhu mapiranja prostora. LIDAR-Lite je postavljen na postolje na kojem su također pričvršćena i 2 servo motora u svrhu rotiranja LIDAR-a oko 2 osi. Koristili smo i arduino ploču koja će služiti kao izvor napajanja servo motora i LIDAR-a te za upravljanje radom motora i LIDAR-a preko serijske komunikacije i arduino razvojne okoline u kojoj ćemo pisati kod. Arduino ploča je povezana USB kabelom sa računalom na koji ćemo slati podatke te ih preko serijske komunikacije primati u processing razvojnu okolinu koju smo koristili za kreiranje skeča za vizualizaciju dobivenih udaljenosti.

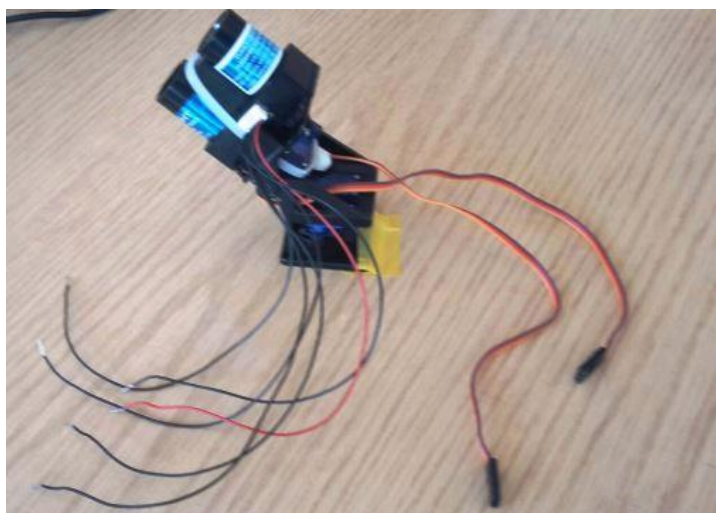
Za razvijanje teme ovog završnog rada potrebne zadatke i koncept možemo podijeliti u 3 koraka:

- 1) Spajanje svih potrebnih dijelova za omogućavanje rada
- 2) Pisanje programskog koda u arduino razvojnu okolinu radi upravljanja LIDAR-om

3) Pisanje programskog koda u processing razvojnu okolinu radi vizualizacije dobivenih podataka

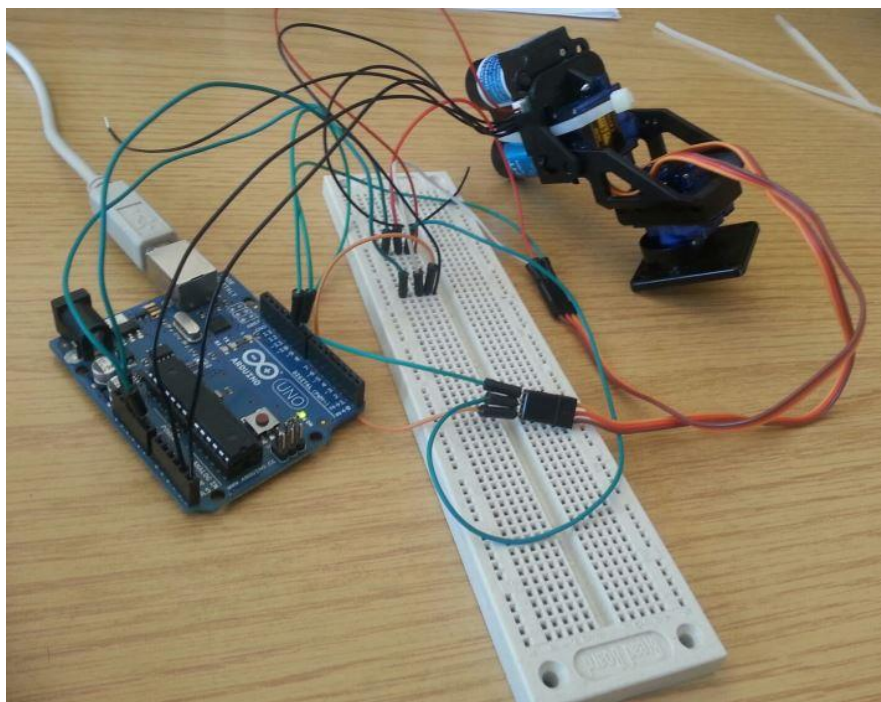
Potrebni dijelovi koji su korišteni u ovom radu su: LIDAR-Lite v2, arduino uno r3, 2 servo sg90 9g motora, postolje (pan-tilt), usb kabel za povezivanje arudina i računala, eksperimentalna pločica i spojne žice.

Postolje se sastoji iz 2 pokretna dijela tako da je moguća rotacija oko osi x i oko osi y. Prvo se trebaju pričvrstiti 2 servo motora na određena mjesta na postolju a zatim i LIDAR-Lite na vrh postolja kao što je prikazano na *slici 5.1*.



*Slika 5.1. LIDAR-Lite i servo motori s postoljem*

Nakon toga trebamo servo motor i LIDAR-Lite spojiti na arduino ploču, ali zbog samo jednog izvora od 5 V na arduino ploči nije moguće spojiti oba motora i LIDAR-Lite. Zbog toga se koristi eksperimentalna ploča koja služi za proširivanje broja priključaka. Koriste se spojne žice kojima ćemo povezati izvor od 5 V i jednu točku eksperimentalne ploče te GND pin arduina sa drugom točkom na eksperimentalnoj ploči. Sada se dovoljan broj uređaja može priključiti na izvor napajanja. Arduino ploča i eksperimentalna ploča sa spojenim LIDAR-om i servo motorima su prikazane na slici ispod.



*Slika 5.2. LIDAR-Lite i servo motori sa arduino pločom*

Zatim nakon što se spoje te 2 ploče pristupa se spajanju servo motora i LIDAR-Lite. Kao što se vidi na slikama servo motori imaju 3 žice ( što je opisano i u poglavlju o hardveru) u ovom primjeru crvenu, crnu i narančastu od kojih crna služi za spajanje na uzemljenje (GND), crvena se spaja na izvor od 5 V i služi za napajanje a narančastu se spaja na digitalni pin arduino ploče te služi za signal pomoću kojeg ćemo upravljati servo motorima. Dok ćemo crvenu i crnu spojiti na već definirane točke na eksperimentalnoj ploči , narančastu od servo motora koji će se rotirati u smjeru osi x spajamo na digitalni pin 9 na arduinu a narančastu od servo motora koji će se rotirati u smjeru osi y spajamo na digitalni pin 10 arduino ploče. LIDAR-Lite na primarnom sučelju J1 ima 6 žica kao što se vidi na slikama od kojih će se koristiti i biti će dovoljne 4. Na slici ispod vidimo značenje svake žice od kojih će se koristiti crvena za napajanje koja će se spojiti na 5 V, zadnja crna žica koja se spaja na uzemljenje (GND) te 2 crne žice koje se nalaze pokraj SDA i SCL koje će služiti za I2C komunikaciju arduina i LIDAR-Lite preko kojih ćemo upravljati njime.



*Slika 5.3. Značenje žica LIDAR-a*

SDA žica se spaja na analogni A4 pin na arduino ploči a SCL na pin A5. Nakon svega što se spojilo što je prikazano na slici ispod , krajnji je dio spajanje USB kabla iz arduino ploče na računalo. Sada kada je sve spojeno kako treba prelazi se na drugi dio pisanje koda u arduino razvojnoj okolini.

Pri pisanju potrebnog koda u arduino razvojnoj okolini koristit će se već opisane 2 potrebne biblioteke za upravljanje servo motorima i I2C komunikacijom. Još jedna vrlo korisna biblioteka koja se može koristiti ali ovdje nije korištena je LIDAR-Lite v2 arduino biblioteka koja je posebno napravljena za LIDAR-Lite. Svrha ove biblioteke je dvostruka:

- 1) Brzi pristup svim općim funkcijama LIDAR-Lite preko arduina
- 2) Kroz ovu biblioteku, korisnici bilo koje platforme će dobiti objašnjenje kako koristiti razne funkcije LIDAR-Lite i vidjeti usput neke arduino primjere.

Prvo što je potrebno su naredbe 'include' koje služe za uključivanje potrebnih biblioteka u arduino skeč. Sintaksa korištenih naredbi je:

#include <Wire.h> - za uključivanje wire biblioteke,

#include <Servo.h> - za uključivanje servo biblioteke;



Dalje definiramo konstante odnosno njihovim vrijednostima pridružujemo imena pomoću koji ćemo ih koristiti u programu. Sintaksa naredbe je:

```
#define imeKonstante vrijednost
```

Konstante koje su definirane u našem programu su:

```
#define LIDARLite_ADDRESS 0x62 – Zadana I2C adresa LIDAR-Lite
```

```
#define RegisterMeasure 0x00 – Komandni registar za zapisivanje vrijednosti za pokretanje mjerenja udaljenosti
```

```
#define MeasureValue 0x04 – Vrijednost koja se upisuje u komandni registar 0x00 za početak mjerenja udaljenosti
```

```
#define RegisterHighLowB 0x8f – Registar za dobivanje i spremanje visokih i niskih bajtova u 1 pozivu
```

Nakon definiranja biblioteka i konstanti koje će se koristiti u programu potrebno je i deklarirati dovoljno varijabli zadavanjem njihovog tipa i vrijednosti. U ovom programu potrebne su 3 varijable koje će biti tipa podataka 'int' (cijelih brojeva), te će biti postavljena na početnu vrijednost 0. To su varijable za pohranu vrijednosti pozicije X servo motora i Y servo motora te varijabla za pohranu izmjerenih udaljenosti. Primjer deklaracije varijable za pohranu udaljenosti je:

```
int distance = 0; gdje je int tip podatka, distance ime varijable i 0 vrijednost.
```

Definira se 2 potrebna servo objekta, varijable tipa servo, koje će služiti za kontroliranje servo motora. Sintaksa definiranja:

```
Servo myXservo; - gdje je Servo tip varijable, a myXservo ime servo motora
```

```
Servo myYservo; - servo objekt za kontrolu Y servo motora.
```

Kao što je rečeno prije arduino skeč se sastoji od 2 glavne funkcije: setup() i loop(). Iduće što pišemo unutar arduino skeča je programski kod unutar setup() funkcije. Setup() funkcija se poziva prilikom pokretanja skeča. Koristi se za inicijaliziranje varijabli, pin modova, početak korištenja biblioteka itd. Ova funkcija će se pokrenuti samo jednom, nakon svakog paljenja ili resetiranja arduino ploče.

U `setup()` funkciji prvo što se definira je serijski izlaz, odnosno funkcija koja služi za otvaranje serijskog porta i razmjenu (slanja) podataka. Sintaksa te funkcije korištene u skeču je:

`Serial.begin(57600);` - gdje je broj u zagradi brzina slanja podataka serijskim portom definirana u bitovima po sekundi (engl. baud rate).

Definirat će se još jedna serijska funkcija koja će ispisivat podatak na serijski port u obliku teksta čitljivog čovjeku, koji će se ispisivat na početku serijskog monitora u arduino razvojnoj okolini. Sintaksa te funkcije je:

`Serial.println(val);` - gdje se unutar zagrada nalazi željeni podatak, u našem slučaju tekst `START` za označavanje početka primanja podataka.

Zatim radi mogućnosti upravljanja servo motorima definiramo funkciju servo biblioteke `attach()` koja služi za povezivanje servo varijabli (motora) na željene pinove na arduino ploči. Sintaksa korištenih funkcija:

`myXservo.attach(10);` - servo motor koji se vrti u smjeru osi x, spaja se na pin 10

`myYservo.attach(10);` - servo motor koji se vrti u smjeru osi y, spaja se na pin 9.

Zadnja korištena funkcija i kod u dijelu `setup()` funkcije je funkcija za upravljanje LIDAR-om, odnosno za pokretanje biblioteke wire i spajanja na I2C sabirnicu kao „master“. Ova funkcija se normalno poziva jednom, te je njena sintaksa:

`Wire.begin();`

Nakon glavne funkcije `setup()` u skeču je definirana funkcija koja će služiti za pisanje potrebnog koda za mjerenje udaljenosti LIDAR-om. Funkcija je nazvana `lidarGetRange` te je tipa cjelobrojne vrijednosti. Prvo napisano u njoj će biti potrebna varijabla tipa cjelobrojne vrijednosti koja će biti postavljena na početnu vrijednost -1.

Slijedi niz funkcija wire biblioteke kojima ćemo upravljati LIDAR-om. Prva od njih je:

`Wire.beginTransmission((int)LIDARLite_ADDRESS);` - koja služi za početak prijenosa I2C komunikacijom na LIDAR-Lite, gdje se u zagradama nalazi zadana adresa. Adresa je 7-bitna od uređaja na koji se šalje.

Zatim imamo funkcije:

`Wire.write((int)RegisterMeasure);` - postavljanje pokazivača registra na 0x00

`Wire.write((int)MeasureValue);` - postavljanje pokazivača registra na adresu 0x00,

koje služe za skupljanje bajtova za slanje od „master“ (arduino) uređaja na „slave“ (LIDAR) uređaj (između poziva za početak slanja i kraj slanja).

Na kraju dolazi funkcija:

`Wire.endTransmission();` - koja služi za kraj prijenosa LIDAR-u i šalje bajtove koji su bili zadržani funkcijom `write()`.

Ovaj niz funkcija u suštini nam služi za slanje podataka I2C komunikacijom LIDAR-u radi početka mjerenja udaljenosti. Nakon toga imam funkciju `delay()` koja služi za čekanje programa određeno vrijeme radi završetka slanja bajtova LIDAR-u.

Nakon nje dolazi opet niz funkcija koje služe za slanje podataka LIDAR-u te postavljanje pokazivača registra na `0x8f` i kraj prijenosa:

```
Wire.beginTransmission((int)LIDARLite_ADRESS);
```

```
Wire.write((int)RegisterHighLowB);
```

```
Wire.endTransmission();
```

Zatim opet definiramo čekanje od 20 ms nakon kojeg će doći funkcija:

```
Wire.requestFrom((int)LIDARLite_ADDRESS, 2);
```

koja služi za zahtijevanje bajtova od LIDAR-a, gdje je `LIDARLite-ADDRESS` adresa a broj 2 u zagradama je količina bajtova koji se zahtijevaju. Čitanjem 2 bajta iz registra `0x8f` dobivamo zapravo izmjerenu udaljenost.

Da bi se radnja čitanja realizirala i napravila kako treba potrebne su još određene funkcije te jedna `if` petlja koja je u ovom slučaju korištena:

```
if(2 <= Wire.available()) // ako su 2 bajta dostupna i primljena ulazi se u if petlju
{
    val = Wire.read(); // prima se viši bajt (prepisuje se preko prošlog čitanja)
    val = val << 8; // „shift-a“ se viši bajt da bude viših 8 bitova
    val |= Wire.read(); // prima se niži bajt kao nižih 8 bitova
}
```

Funkcija `Wire.available()` služi za vraćanje broja dostupnih bajtova za daljnje dohvaćanje sa funkcijom `read()`. Ova funkcija se poziva na „slave“ uređaju (arduino) nakon

poziva funkcije requestFrom(). Funkcija Wire.read() čita bajtove koji su poslani sa LIDAR-a na arduino. Nakon ispunjenog if uvjeta i izvršavanja naredbi unutar if petlje podatak koji se vraća je varijabla val odnosno izmjerena udaljenost.

Dalje u arduino skeču napisana je funkcija koja prima 3 argumenta cjelobrojnih vrijednosti te služi za ispisivanje dobivene udaljenosti i pozicije servo motora na serijski port, koje možemo čitati u serijskom monitoru. Ta funkcija izgleda ovako:

```
void serialPrintRange(int posX,int posY, int distance)
{
    Serial.print(posX);
    Serial.print(",");
    Serial.print(posY);
    Serial.print(",");
    Serial.println(distance);
}
```

Zadnja te jedna od glavnih funkcija koja je potrebna u svakom arduino skeču koja će biti ukratko opisana je funkcija void loop() koja se uzastopno vrti u petlji te za razliku od funkcije setup() ova funkcija se ponavlja beskonačan broj puta sve dok se arduino ploča ne isključi.

U ovoj funkciji se nalaze 2 glavne for petlje te još 4 for podpetlje, gdje će 1 glavna for petlja služiti za rotiranje X servo motora u jednu stranu dok će 2 for petlje koje se nalaze unutar te služiti za rotiranje Y servo motora prvo u jednu stranu zatim nazad u početni položaj. A 2 glavna for petlja će raditi identično osim što će se X servo motor vraćati u početni položaj.

Opisat će se dio koda kojim će se objasniti proces, ostatak koda radi vrlo slično pa neće biti potrebe za opisivanjem cijeloga. Dio koda u funkciji void loop():

```
for(posX = 60; posX <= 100; posX += 1) // zakretanje X osi
{
    for(posY = 100; posY <= 180; posY += 1) { // zakretanje Y osi
        myXservo.write(posX);
```

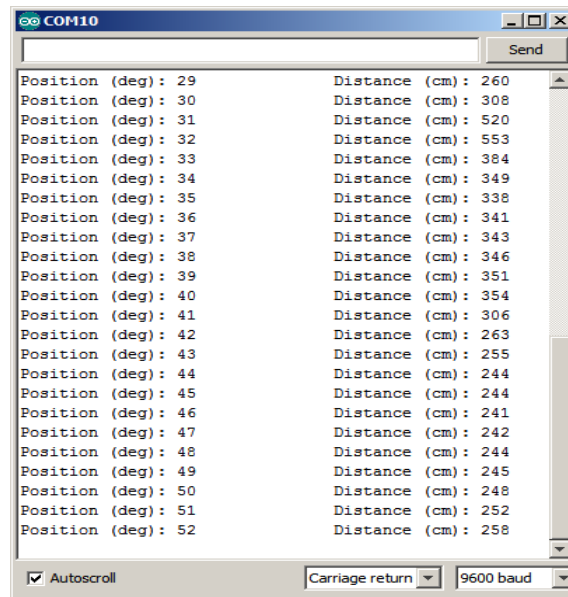
```
myYservo.write(posY);  
  
distance = lidarGetRange();  
  
serialPrintRange(posX, posY, distance); } ...
```

Prva for petlja će služiti za kretanje X servo motora od početnog položaja koji u ovom slučaju nije 0° već 60° prema krajnjem dovoljnom položaju od 100°. Druga for petlja služi za rotiranje Y servo motora iz početnog položaja od 100° do krajnjeg položaja 180°. Kod je napisan tako da za svaki 1 stupanj X servo motora se ulazi u drugu for petlju u kojoj se Y servo motor pomjera od 100 do 180 stupnjeva, zatim se pomjera od 180 pa do 100. Tek tada se X servo motor pomjeri opet za 1 stupanj a Y servo motor napravi pomak iz početnog u krajnji položaj i iz krajnjeg početni položaj. Radnja završava tek kad se X servo motor pomjeri svih 40 stupnjeva pri čemu će se za svaki pomjeren 1 stupanj X servo motora, Y servo motora pomjerit za 80 stupnjeva u oba smjera.

Unutar petlje vidimo poziv dvije funkcije. Za svaki položaj u kojem se nađu servo motori pozivaju se 2 funkcije koje smo napisali u programu: funkcija `lidarGetRange()` kojom se dobiva izmjerena udaljenost te funkcija `serialPrintRange` kojom ispisujemo te vrijednosti na serijski port.

Time je završeno opisivanje cijelog potrebnog arduino koda koji je napisan za upravljanje LIDAR-om i servo motorima. Kao što se vidi arduino skeč nije složen i 2 glavne značajke su dobivanje informacija I2C protokolom od LIDAR-a i upravljanje njima te upravljanje servo motorima. Nakon toga imamo program koji odlično upravlja cijelim sklopovljem i omogućuje mjerenje udaljenosti i čitanje istih. Sada je potrebno još te vrijednosti pretvoriti u mapu (sliku) odnosno vizualizirati podatke. To će biti obavljeno pomoću processing skeča opisanog u idućem koraku.

Primjer ispisa izmjerenih udaljenosti u odnosu na položaj jednog motora u arduino serijskom monitoru je prikazan na slici ispod.



*Slika 5.4. Ispis pozicije i udaljenosti na serijskom monitoru*

Zadnji i 3 korak ovog završnog rada je vizualizacija dobivenih udaljenosti koju ćemo postići koristeći vrlo jednostavan i odličan program za grafički prikaz raznih ideja po imenu Processing. Mogućnosti i processing razvojna okolina je već ranije opisana u poglavljima pa će se sad objasniti potreban kod pisan processing skeču za vizualizaciju izmjenjenih udaljenosti u cilju stvaranja mape prostora.

Prvi potrebni korak je omogućiti processingu čitanje podataka potrebnih za stvaranje slike prostora, a ti podaci se čitaju sa serijskog porta na kojem se nalaze i učitavaju se u processing skeč idućom naredbom koja je na početku processing skeča:

```
import processing.serial.*;
```

Zatim deklariramo 3 potrebne varijable u koje ćemo spremati vrijednost x koordinate, vrijednost y koordinate te vrijednost boje točke na toj koordinati. Sve tri su tipa float brojevi sa decimalama.

```
float pcolor = 0;
```

```
float xVal = 0;
```

```
float yVal = 0;
```

Kreiramo i jedan objekt klase Serial koja služi za slanje i primanje podataka koristeći serijski komunikacijski protokol. Nazvat ćemo ga myPort.

Serial myPort;

Processing skeč se sastoji od 3 glavne funkcije. Prva od njih je setup() funkcija koja je samo jedna u svakom skeču te se pokreće samo jednom i služi za inicijaliziranje osnovnih stvari kao što su dimenzije prozora za grafički prikaz, za učitavanje slika i dr. Ispod je kod naše setup() funkcije koji ćemo objasniti:

```
void setup() {  
  size(400,200);  
  
  printArray(Serial.list()); // lista svih slobodnih serijskih portova  
  
  myPort = new Serial(this, Serial.list()[0], 57600); // otvaranje porta koji se koristi pri baud-  
  rate od 57600  
  
  myPort.bufferUntil('\n'); // ne generirat serialEvent() ako nismo dobili znak u novoj liniji  
  
  background(0);  
}
```

Funkcija size() se koristi za dimenzioniranje prozora za grafički prikaz gdje je prvi broj u zagradi širina u pikselima a drugi visina prozora u pikselima. Ova funkcija se može samo jednom napisati i mora biti na početku koda unutar setup() funkcije.

printArray(Serial.list()); je funkcija za ispisivanje liste svih slobodnih serijskih portova.

Iduća linija koda postavlja pronađeni slobodni port adrese 0 kao novi serijski objekt klase Serial nazvan myPort, te otvara taj serijski port koji se koristi pri čitanju podataka brzinom od 57600 bita po sekundi. Ova brzina mora biti identična kao i ona korištena u arduino skeču da bi se podatak mogao pročitati.

Iduća linija myPort.bufferUntil('\n') ne dopušta da se generira serialEvent() ako se nije dobio znak u novoj liniji ( podaci se spremaju dok ne dođe znak u novoj liniji te se onda obrađuju).

Background(0) je funkcija za biranje boje pozadine prozora u ovom slučaju crne, jer broj u zagradi predstavlja rgb vrijednost a 0 pripada crnoj.

Iduća od 3 funkcije je funkcija draw() koja može samo jedan biti u skeču te se nalazi nakon funkcije setup(). Unutar nje kod se izvršava uzastopno sve dok se ne prekine funkcijom noLoop() ili dok program ne stane.

```
void draw() {
    stroke(pcolor);
    point(xVal,yVal);
}
```

Stroke() funkcija služi za postavljanje boje za crtanje linija i granica oko različitih oblika. U ovom slučaju njome postavljamo boju točki sa pcolor vrijednostima koje će biti u rasponu od 0 do 255 sivi tonovi.

Funkcija point() služi za crtanje točki određenih koordinata u prostoru sa dimenzijom jednog piksela. Prva vrijednost u zagradi xVal je horizontalna vrijednost točke a druga je okomita vrijednost točke.

Zadnja i 3 funkcija koja je korištena u processing skeču je void SerialEvent() u kojoj ćemo napisati potrebni kod za ispravno čitanje traženih vrijednosti odnosno stringova uz određene uvjete, te boju pcolor i način na koji će se crtati u draw funkciji. Funkcija koja je korištena se vidi ispod i bit će detaljno objašnjen kod unutar nje.

```
void serialEvent(Serial myPort) {
    String inString = myPort.readStringUntil('\n');
    if (inString !=null) {
        inString = trim(inString);
        float[] colors = float(split(inString, ","));

        if(colors.length >=3) {
            xVal = colors[0];
            yVal = colors[1];
            pcolor = map(colors[2], 0, 400, 255,0); } } }
```



Funkcija `serialEvent()` poziva se kada su podaci dostupni, te ima unutar zagrade argument koji prima a to je objekt `myPort` klase `Serial` preko kojega se primaju novi podatci. Prva linija koda unutar funkcije `serialEvent` kreiranje objekta `inString` klase `String` koja dopušta manipuliranje stringovima teksta nego što to dopuštaju znakovni nizovi. Moguće su razne uporabe s klasom `String` kao što traženje te mijenja stringova, mijenjanje podstringova i mnogo više. U prvoj liniji koda primaju se i čitaju podatci odnosno string podataka jednog reda i taj string se sprema u objekt `inString`. Kao što se vidi za svaku red primljenih podataka ući se ponovo u 2 if petlje koje će taj string obrađivati.

U prvoj if petlji ima uvjet koji govori da se u petlju ulazi samo ako `inString` nije prazne vrijednosti odnosno ako imamo spremljen dobiveni string u njemu. Dalje funkcija `trim()` služi za uklanjanje „bijelog prostora“ s početka i kraja stringa.

U idućoj liniji definira se varijabla `colors` koja je deklarirana kao niz znakova tipa `float` ( brojevi sa decimalnom točkom). Funkcija koju vidimo u toj liniji koda `split()` služi za razdvajanje stringa u dijelove koristeći neki znak kao granični. Taj znak određuje znak ili znakove koji označavaju granicu između svakog dijela stringa.

U ovom slučaju razdvaja se string `inString` na mjestima gdje se nalazi zarez jer podaci dolaze u obliku: pozicija Xserva, pozicija Yserva, udaljenost. Razdvaja se taj string u podstringove gdje će nakon toga `inString[0]` biti pozicija X serva, `inString[1]` pozicija Yserva i na 3 adresi će biti udaljenost. Zatim se ti podstringovi pretvaraju u niz brojeva tipa `float` ( brojeve sa decimalnom točkom).

Nakon toga dolazi iduća if petlja čiji je uvjet da dužina niza `colors` mora biti jednaka 3 ili veća odnosno da primi sva tri potrebna podatka. Ako je tako onda se ulazi u nju i piše kod koji će služiti za mapiranje tih vrijednosti.

Prvi član niza `colors[0]` je pozicija X serva i sprema se u varijablu `xVal`, a drugi član je pozicija Y serva i sprema se u `yVal` varijablu koje će služiti kao koordinate za crtanje točki. Zadnja linija koda u 2 if petlji a i cijelom skeču je naredba za mapiranje prostora pomoću udaljenosti u rasponu sivih tonova od 0 do 255.

```
pcolor = map(colors[2], 0, 400, 255,0);
```

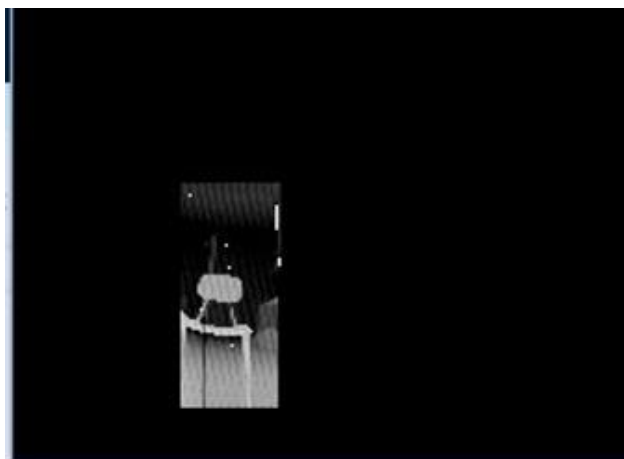
Funkcija `map()` služi za pretvaranje vrijednosti brojeva iz jednog raspona brojeva u drugi novi raspon vrijednosti. Ovom funkcijom zajedno sa koordinatama točaka ćemo izvršiti mapiranje. Kao što se vidi funkcija `map()` unutar zagrade ima 5 argumenata gdje je prvi argument `colors[2]` 3 član niza i predstavlja udaljenost i to je vrijednost koja će se konvertirati

iz raspona vrijednosti od 0 do 400 što su nam 2 i 3 argument a to su donja i gornja granica prvog raspona, u raspon novih vrijednosti od 255 do 0 a to su 4 i 5 argument koji predstavljaju novu donju i gornju granicu raspona. Raspon od 0 do 255 predstavlja sve sive tonove a u ovom slučaju taj raspon je invertiran radi boljeg i jasnijeg mapiranja. Zatim nakon ove funkcije koja se sprema u varijablu pcolor draw() funkcija sa stroke() i point() će izvršiti krajnje crtanje tih vrijednosti. Time je završen cijel kod processing skeča potreban za mapiranje i na slikama ispod su prikazani rezultati.



*Slika 5.5. Grafički prikaz stolice*

Na slici 5.5. brzina okretanja servo motora je bila veća za 1 stupanj pa je kvaliteta dobivene slike manja nego na slici 5.6. gdje uočavamo dosta bolju kvalitetu i jasniju sliku jer smo za nju koristili najmanje rotiranje servo motora, zbog čega su podaci udaljenosti utvrđeniji.



*Slika 5.6. Grafički prikaz stolice sa manjom brzinom motora*

Na slikama se vidi da svaka točka zauzima jedan piksel te da je mapiranje izvršeno u sivim tonovima od 255 do 0 tako da predmeti koji su manje udaljeni su svjetlije boje (bije), a predmeti koji su udaljeniji su sve tamnije boje pa sve do crne koji su najdalji. Time je jednostavno i jasno izvršeno mapiranje i slike su dovoljne kvalitete za raspoznavanje predmeta i mape prostora.

## 6. ZAKLJUČAK

S obzirom na dobivene podatke udaljenosti izmjerene uređajem LIDAR-Lite v2 te dobivenu mapu prostora nakon obrađivanja podataka u Processing razvojnoj okolini može se zaključiti sljedeće:

LIDAR-Lite v2 je sposoban vršiti precizna mjerenja udaljenosti na manjim udaljenostima te što su predmeti u prostori veći i što ih manje ima to su mape prostora preglednije, a ako imamo dosta predmeta na raznim udaljenostima različitih veličina dobivene slike neće biti jako pregledne. Međutim LIDAR-Lite v2 dovoljno precizno mjeri udaljenosti na daljinama maksimalno do 40 metara, te što je brzina servo motora manja time se dobiva više podataka za iste točke prostora samim time i mapa prostora je dosta kvalitetnija.

Postupak je spojiti LIDAR-lite v2 te servo motore na arduino uno ploču koja će im davati signale te upravljati njima. Nakon spojenih servo motora i LIDAR-Lite-a na arduino ploču, ploču uključujemo USB kabelom na računalo. Pokreće se arduino skeč koji je napisan da upravlja LIDAR-om preko I2C komunikacije te prima izmjerene udaljenosti i da pokreće servo motore u određenom rasponu stupnjeva određenom brzinom. Nakon primljenih podataka preko serijske komunikacije učitavamo ih u processing skeč koji će ih obrađivati i vršiti određene naredbe kojima će izvršiti vizualizaciju primljenih podataka te stvoriti mapu željenog prostora.

Ovo je samo jedna od primjena LIDAR-a, osim nje LIDAR se koristi na svim robotima radi izbjegavanja predmeta na koje nailaze te upravljanja u prostoru. Ipak ovaj uređaj se ne koristi u primjenama mapiranja prostora jer postoje mnogo moćniji te složeniji LIDAR-i koji mogu mjeriti udaljenosti da mnogo većim dometima te pomoću kojih se dobiva jako kvalitetna 3D mapa prostora.

## LITERATURA

- [1] S Interneta, [https://en.wikipedia.org/wiki/Lidar#cite\\_note-Oxford-2](https://en.wikipedia.org/wiki/Lidar#cite_note-Oxford-2),  
23. kolovoza 2016.
- [2] „LIDAR-Lite v2“, s Interneta, <https://www.sparkfun.com/products/retired/13680>,  
24. kolovoza 2016.
- [3] „PulsedLight3D/LIDAR-Lite-Documentation“,  
s Interneta, <https://github.com/PulsedLight3D/LIDAR-Lite-Documentation>,  
24. kolovoza 2016.
- [4] S Interneta, <https://www.arduino.cc/en/Guide/Introduction>, 29. kolovoza 2016.
- [5] S Interneta, <https://en.wikipedia.org/wiki/Arduino>, 29. kolovoza 2016.
- [6] S Interneta, <https://www.arduino.cc/en/Main/ArduinoBoardUno>, 29. kolovoza 2016.
- [7] „SG90 9g Micro Servo“, s interneta <http://www.micropik.com/PDF/SG90Servo.pdf>,  
30. kolovoza 2016.
- [8] „Arduino Software (IDE)“, s Interneta,  
<https://www.arduino.cc/en/Guide/Environment>, 30. kolovoza 2016.
- [9] S Interneta, <https://www.arduino.cc/en/Reference/Libraries>, 3. rujna 2016.
- [10] SInterneta,  
[https://en.wikipedia.org/wiki/Processing\\_\(programming\\_language\)#Wiring,2C\\_Arduino.2C\\_and\\_Fritzing](https://en.wikipedia.org/wiki/Processing_(programming_language)#Wiring,2C_Arduino.2C_and_Fritzing), 1. rujna 2016.
- [11] S Interneta, <https://processing.org/reference/environment/>, 3. rujna 2016.
- [12] S Interneta, <https://www.arduino.cc/en/Reference/HomePage>, 6. rujna 2016.
- [13] S Interneta, <https://processing.org/reference/>, 6. rujna 2016.

## POPIS OZNAKA I KRATICA

APD	lavinska fotodioda
AC	izmjenična struja
DC	istosmjerna struja
engl.	engleski
EEPROM	električno izbrisiva programibilna ispisna memorija
GND	uzemljenje
I2C	sučelje za komunikaciju, među-integrirani krug
ICSP	unutar strujnog kruga serijsko programiranje
IDE	integrirana razvojna okolina
LED	svjetleća dioda
LIDAR	svjetlosna detekcija i mjerenje dometa
PCB	printana pločica strujnih krugova
PDE	processing razvojna okolina
PWM	pulsno širinska modulacija
SCL	serijski takt I2C protokola
SDA	serijski podaci I2C protokola
SoC	sustav-na-čipu tehnologija
SPC	jezgra procesiranja signala
SPI	serijsko periferno sučelje (komunikacijsko)
SRAM	statička memorija sa nasumičnim pristupom
TWI	dvo-žično sučelje
USB	univerzalna serijska sabirnica
VCSEL	površinsko-emitirajući laser sa vertikalnim šupljinama

## SAŽETAK

LIDAR je optički mjerni instrument koji mjeri udaljenost osvjetljavajući metu laserskim zrakama. LIDAR koristi ultraljubičasto, vidljivo ili skoro infracrveno svjetlo za stvaranje slike objekta. LIDAR-Lite v2 je idealan za korištenje u dronovima, robotima ili u bezposadnim vozilima u situacijama kada je potreban pouzdan i moćan senzor blizine (udaljenosti) a da ne zauzima previše prostora. Za komuniciranjem s ovim senzorom potrebno je standardno I2C sučelje ili PWM.

Jedan od načina spajanja koji se koristi je I2C umrežavanje. LIDAR-Lite v2 spajamo na mikrokontrolersku ploču arduino uno koja je prilagođena i sposobna za upravljanje funkcijama LIDAR-a. Jedna od primjena LIDAR-a je mapiranje prostora pomoću izmjerenih udaljenosti. Potrebno je spojiti LIDAR na pravilan način na arduino ploču te smjestiti ga na postolje pogonjeno servo motorima pomoću kojih će se vršiti mjerenje udaljenosti različitih točaka u prostoru. Spajaju se servo motori na napajanje te na arduino ploču koja će upravljati njima pomoću PWM. Arduino i LIDAR komuniciraju serijskom I2C komunikacijom pomoću SDA i SCL pinova. LIDAR-Lite će mjeriti udaljenosti u prostoru koje će arduino primiti te ispisivati na serijskom monitoru.

Za definiranje rada arduina i načinom na koji upravlja LIDAR-om te svim potrebnim funkcijama i naredbama koje su potrebne za pravilan rad LIDAR-a te prijenosa i spremanja podataka koristi se arduino razvojna okolina te napisani programski kod unutar arduino skeča. Za vizualizaciju podataka i stvaranje mape prostora koristi se processing razvojna okolina u kojoj se definira i piše potrebni kod kojim će se primiti i obrađivati podaci iz arduina dobiveni preko serijskog porta. Processing skečom se stvara mapa prostora dobivena pomoću izmjerenih udaljenosti te se jasno može vidjeti predmete koji su se našli u toj prostoriji.

Iako LIDAR-Lite jako brzo i precizno mjeri udaljenosti te se one vizualiziraju u 2D mapu ipak njegova najveća primjena nije to jer postoje mnogo bolji LIDAR senzori za stvaranje 3D mapa na jako velikim udaljenostima. Ali naravno LIDAR-Lite se koristi jako mnogo radi brzog mjerenja i malih dimenzija u zadacima gdje mapiranje i nije toliko bitno.