

POROČILO »ZOPp«

Simon LIPOVŠEK

Razred: E3.C

Šolsko leto 2021/2022

Datum: 12.10.2021

Kazalo:

[1. Blok shema 3](#_Toc100572933)

[2. Delovanje 3](#_Toc100572934)

[3. Simulacija 5](#_Toc100572935)

[4. Vezalna shema 5](#_Toc100572936)

[5. PCB 6](#_Toc100572937)

[6. Kosovnica 7](#_Toc100572938)

[7. Kritični material 7](#_Toc100572939)

[8. Koda 8](#_Toc100572940)

[3. Komentar 12](#_Toc100572941)

# Blok shema

Diagram

Description automatically generated

# Delovanje

Pred uporabo je potrebno vklopiti napravo v napajanje. Sedaj se nam prižge zaslon, na katerem je potrebno izbrati natančnost senzorja. S tem izberemo premik v z in zasuk plošče. Če izberemo najbolj natančen način bo sekiral dlje časa in tudi naredil bolj natančno skico.

Delovanje se začne ko vklopimo napravo v napajalni vir. Tedaj naprava preveri če je SD kartica vstavljena in nadaljuje če je. Zatem naprava začenja vrteti podlago in vertikalno os po prej določenem programu, in sicer najprej izmeri razdaljo, jo preračuna in shrani na SD kartico. Podlaga se sedaj zavrti za eno stopinjo in proces merjenja se ponovi. Ko se podlaga zavrti za 360 stopinj se senzor na z osi premakne za 1mm navpično navzgor in podlaga ponovno začne proces merjenja in shranjevanja. Ko je skeniranje objekta končano podatke prenesemo v računalnik in tam uporabimo program Mesheye za tvorjenje 3D objekta. Ta program nam tudi omogoča pretvarjanje točkovnega oblaka v stl file katerega lahko kasneje urejamo v skoraj kateremkoli 3d modelirnem programu.

1. **ARDUINO PLOŠČA**

Arduino je odprto-kodni programski jezik z izvorom v programskem jeziku C. Arduino plošča je naprava na kateri lahko programiraš čip ATMEGA32, lahko pa uporabljaš celotno ploščo ter njene izhodne pine za povezovanje z zunanjimi komponentami. Plošča vsebuje tudi flash pomnilnik in komunikacijski procesor za povezavo z računalnikom. Plošča nam omogoča programiranje čipa z USB konektor. Preko USB vmesnika lahko Arduino ploščo napajamo, programiramo in z njo komuniciramo. Arduino plošča podpira analogne , digitalne vhodne/izhodne napetosti in PWM. Analogni signali so shranjeni kot 10 bitne informacije in obratno ima izhodna analogna zapis 1024 stanj. Omogoča tudi različne komunikacijske protokole z napravami, kot sta: \_\_, I2C.

**Uporaba v projektu**

Arduino plošča bo zbirala podatke iz različnih komponent in podajala ukaze drugim napravam.

1. MODUL ZA SD KARTICO

Modul za SD kartico se uporablja za dolgoročno oz. kratkoročno shranjevanje podatkov. Ta modul se uporablja za mikro SD kartice. Za napajanje potrebuje 4 do 5 voltov in od 0,2 do 200 mA. Z modulom lahko komuniciramo preko SPI protokola. Hitrost prenosa podaktov je od 0 do 120 KB na sekundo.

**Uporaba v projektu**

Modul za SD kartico se uporablja kot prostor za shranjevanje x, y, z točka na površini objekta. Te podatke kasneje prenesemo na računalnik kjer s pomočjo programa tvorimo 3D model objekta.

1. LASERSKI MERILNIK

Laserski senzor s pomočjo laserja izmeri oddaljenost predmeta. Oddaljenost izračuna z časovno razliko oddanega signala in sprejetja odboja tega signala.

**Uporaba v projektu**

Naloga laserskega merilnika je da izmeri oddaljenost od senzorja do predmeta položenega na vrteči plošči. Meritev se opravi na vsake 2mm zunanjega obsega vrteče podlage.

1. KORAČNI MOTOR

Koračni motor je elektromehanski del, ki pretvori električni signal v mehansko rotacijo grede. Vsak pomik osi za točno določen kot se izvede ob električnem pulzu. Koračni motor torej pretvarja električne pulze v rotacijo njegove grede.

**Uporaba v projektu**

V projektu uporabljam dva koračna motorja. Prvi koračni motor vrti podlago in spreminja x in y pozicijo objekta. Drugi koračni motor pa premika Z pozicijo laserskega merilnika. Poleg razdalje zmerjene z laserskim merilnikom se pošljejo tudi x, y, z koordinate te meritve.

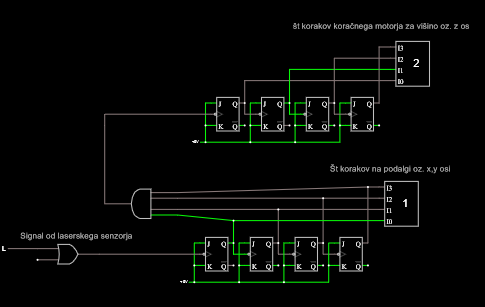
1. NAPAJANJE

Za napajanje uporabljamo napajalni kabel in adapter. Napajalni adapter je vključen v vtičnico. Znotraj adapterja je električno vezje, ki pretvori 230V izmenično napetost v 12V enosmerno napetost.

**Uporaba v projektu**

Za napajanje je uporabljen 12V kabel. S tim direktno napajamo motorje in regulator napetosti. Regulator napetosti zniža 12V v 5V in s tem napaja Arduino ploščo in druge 5V module.

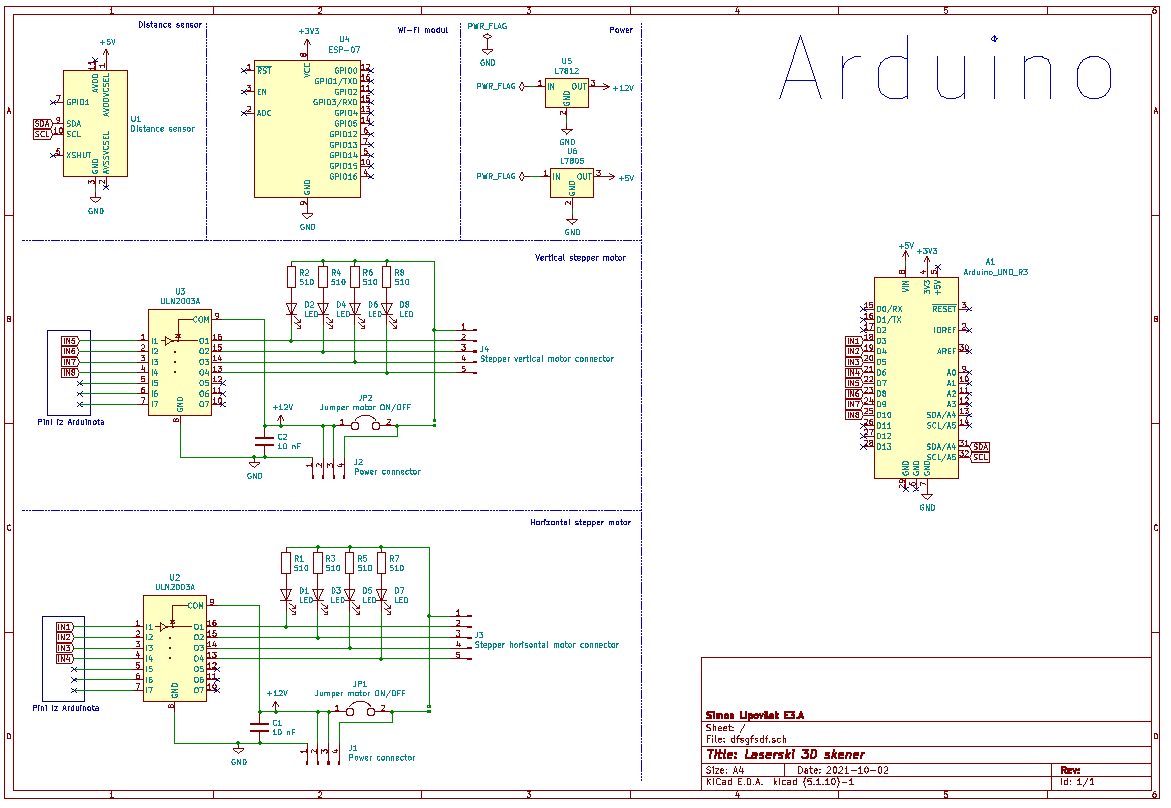
# Simulacija



Simulacija je bila izvedena s programom Circuit simulator. Delujoče vezje je na linku: <https://tinyurl.com/yg5t72ye>.

Simulacija prikazuje delovanje servomotorjev znotraj programa. Na levi strani simulacije lahko vidimo ali vrata s katerimi sporočimo vezju da je merilnik razdalje zaključil merjenje in čaka da spremeni pozicijo. Signal merilnika razdalje premakne koračni motor na x,y osi. Kot s katerim je koračni motor premaknjen na x,y osi je prikazan s številko na spodnjem desnem zaslonu. Ko je x,y osni koračni motor naredil cel krog, pošlje signal vertikalnemu koračnemu motorju. Ta resetira zaslon na x,y koračnem motorju in izpiše svoj zavrten kot. Kot s katerim se je vertikalni motor zavrtel je prikazan na zgornjem desnem zaslonu.

# Vezalna shema



Koračna motorja in laserski merilnik ne bodo neposredno pritrjene na vezje ampak bodo pri spajkani preko pinov na vezje.

# PCB

Pri vezavi so bila podnožja nekaterih komponent zamenjana z podnožjem njim najbolj podobnim. sss

**Primer vezave v KiCad programu:**

Vezave prvega sloja: Pogled vezav drugega sloja:

Diagram, schematic

Description automatically generated

**3D predstava vezja v KiCad programu**

3d pogled z ptičje perspektive:

Diagram

Description automatically generated

# Kosovnica

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Vrednost** | **Raster** | **Kos** | **Opomba** |
| **Upor** | 510 | 7,5 | 8 |  |
| **LED** |  |  | 8 |  |
| **ULN2003A** |  |  | 2 |  |
| **Jumper** |  |  | 2 |  |
| **Napajalni konektorji** |  |  | 70 | Uporaba za pripenjanje arduina in drugih odmaknjenih komponet |
| **Kondenzator** | 10 nF |  | 2 |  |
| **L7812** | 12V |  | 1 |  |
| **L7805** | 5V |  | 1 |  |
| **Wi-Fi modul** |  |  | 1 |  |
| **Laserski senzor** |  |  | 1 |  |
| **Arduino uno** |  |  | 1 |  |

# Kritični material

Za projekt je potrebno imeti: laserski merilnik, 2 koračna motorja, Arduino. Za upravljanje motorjev je potrebni uravnalnik napetosti saj je potrebno dvigniti na 12V.

# Koda

Koda se deli na dva dela, prvi del se izvede v arduinu med tem ko drugi del pa na računalniku. S tem prihranimo na času pri samem skeniranju in pridobimo lepše in bolj elegantne predstave podatkov oz. objekta.

**Koda za Arduino**

**#include <Arduino.h>**

**//SD**

**#include <SPI.h>**

**#include <SD.h>**

**File myFile;**

**// include the library code:**

**#include <LiquidCrystal.h>**

**// initialize the library by associating any needed LCD interface pin**

**// with the arduino pin number it is connected to**

**const int rs = 8, en = 7, d4 = 5, d5 = 4, d6 = 3, d7 = 2;**

**LiquidCrystal lcd(rs, en, d4, d5, d6, d7);**

**//laser**

**#include <Wire.h>**

**#include "Adafruit\_VL6180X.h"**

**Adafruit\_VL6180X vl = Adafruit\_VL6180X();**

**// Define stepper motor connections and steps per revolution:**

**#define dirPin1 A2**

**#define stepPin1 A3**

**#define dirPin2 6**

**#define stepPin2 9**

**//zaznavannje napake**

**int motor\_upocasnitev = 500;**

**int ponovitve = 500;**

**int visina\_z\_osi = 0.5; // 2**

**int vrtljaj\_z\_osi = 1600 \* visina\_z\_osi; //3200**

**bool napaka = false;**

**float razdalja = 0; //uvedba spremenjlivke zaradi unkcije**

**float x = 0;**

**float y = 0;**

**float z = 0;**

**void setup() {**

**// Declare pins as output:**

**pinMode(stepPin1, OUTPUT);**

**pinMode(dirPin1, OUTPUT);**

**pinMode(stepPin2, OUTPUT);**

**pinMode(dirPin2, OUTPUT);**

**lcd.begin(16, 2);**

**// Print a message to the LCD.**

**lcd.print("hello, world!");**

**// Open serial communications and wait for port to open:**

**Serial.begin(115200);**

**while (!Serial) {;}**

**//Serial.print("Initializing SD card...");**

**if (!SD.begin(10)) {**

**Serial.println("initialization failed!");**

**while (1);**

**}**

**//Serial.println("initialization done.");**

**while (!Serial) {**

**delay(1);**

**}**

**Serial.println("Adafruit VL6180x test!");**

**if (! vl.begin()) {**

**Serial.println("Failed to find sensor");**

**while (1);**

**}**

**Serial.println("Sensor found!");**

**// open the file. note that only one file can be open at a time,**

**// so you have to close this one before opening another.**

**myFile = SD.open("test.txt", FILE\_WRITE);**

**// if the file opened okay, write to it:**

**if (myFile) {**

**Serial.print("Writing to test.txt...");**

**myFile.println("This is a test file :)");**

**// close the file:**

**myFile.close();**

**Serial.println("done.");**

**} else {**

**// if the file didn't open, print an error:**

**Serial.println("error opening test.txt");**

**}**

**}**

**void errors(float status){**

**napaka = true;**

**if ((status >= VL6180X\_ERROR\_SYSERR\_1) && (status <= VL6180X\_ERROR\_SYSERR\_5)) {**

**Serial.println("System error");**

**}**

**else if (status == VL6180X\_ERROR\_ECEFAIL) {**

**Serial.println("ECE failure");**

**}**

**else if (status == VL6180X\_ERROR\_NOCONVERGE) {**

**Serial.println("No convergence");**

**}**

**else if (status == VL6180X\_ERROR\_RANGEIGNORE) {**

**Serial.println("Ignoring range");**

**}**

**else if (status == VL6180X\_ERROR\_SNR) {**

**Serial.println("Signal/Noise error");**

**}**

**else if (status == VL6180X\_ERROR\_RAWUFLOW) {**

**Serial.println("Raw reading underflow");**

**}**

**else if (status == VL6180X\_ERROR\_RAWOFLOW) {**

**Serial.println("Raw reading overflow");**

**}**

**else if (status == VL6180X\_ERROR\_RANGEUFLOW) {**

**Serial.println("Range reading underflow");**

**}**

**else if (status == VL6180X\_ERROR\_RANGEOFLOW) {**

**Serial.println("Range reading overflow");**

**}**

**delay(50);**

**}**

**void kordinate(int i, int j, float range){**

**int razdalja = 100-range;**

**float kot = 3.1415926535897932384626433832795\*2/3200 \* j; //najmanjši kot ki ga lahko motor naredi s korakom in št korakov**

**x = cos(kot) \* razdalja; //kotna funkcija za izračun x kordinate**

**y = sin(kot) \* razdalja; //kotna funkcija za izračun y kordinate**

**z = i \* visina\_z\_osi; // druga številka je za koliko se z os vzdigne ko se zavrti za en obrat**

**}**

**float laser(float naprava){**

**/\* float lux = vl.readLux(naprava);**

**Serial.print("Lux: "); Serial.println(lux);**

**\*/**

**uint8\_t range = vl.readRange();**

**uint8\_t status = vl.readRangeStatus();**

**if (status == VL6180X\_ERROR\_NONE) {**

**//Serial.print("Range: "); Serial.println(range);**

**// return range; //lux;**

**razdalja = range;**

**}else{// Some error occurred, print it out!**

**errors(status);}**

**return razdalja;**

**}**

**void sd(float x, float y, float z){**

**myFile = SD.open("test.txt", FILE\_WRITE);**

**// if the file opened okay, write to it:**

**if (myFile) {**

**Serial.print("Writing to test.txt...");**

**myFile.print(x);**

**myFile.print(";");**

**myFile.print(y);**

**myFile.print(";");**

**myFile.println(z);**

**// close the file:**

**myFile.close();**

**Serial.println("done.");**

**} else {**

**// if the file didn't open, print an error:**

**Serial.println("error opening test.txt");**

**}**

**}**

**void loop() {**

**digitalWrite(dirPin2, LOW); // Set the spinning direction clockwise:**

**digitalWrite(dirPin1, LOW); // Smer vrtenja**

**// Set the spinning direction counterclockwise:**

**for ( int i = 0; i < ponovitve; i++){ // Kolikokrat se bo z os zavrtela za 360°**

**for (int j = 0; j < 3200; j++){ //vrtenje podlage za 2pi**

**digitalWrite(stepPin2, HIGH);**

**delayMicroseconds(motor\_upocasnitev);**

**digitalWrite(stepPin2, LOW);**

**delayMicroseconds(motor\_upocasnitev);**

**if (!napaka){**

**float range = laser(VL6180X\_ALS\_GAIN\_5);**

**//Serial.println(range);**

**kordinate(i, j, range);**

**Serial.println(x);**

**Serial.println(y);**

**Serial.println(z);**

**sd(x,y,z);**

**}else{**

**napaka = false; //Če je napaka jo ignorera in ne zapiše nič**

**}**

**}**

**for (int k = 0; k < vrtljaj\_z\_osi; k++){ //En vrtljaj na osi oz, laho se spremeni na kaj druzga 3200**

**digitalWrite(stepPin1, HIGH);**

**delayMicroseconds(motor\_upocasnitev);**

**digitalWrite(stepPin1, LOW);**

**delayMicroseconds(motor\_upocasnitev);**

**}**

**}**

**digitalWrite(dirPin1, HIGH);**

**// Set the spinning direction counterclockwise:**

**for ( int i = 0; i < ponovitve; i++){ // Kolikokrat se bo z os zavrtela za 360° in postavitev senzorja v začetno pozicijo**

**for (int k = 0; k <3200; k++){ //En vrtljaj na osi oz, laho se spremeni na kaj druzga**

**digitalWrite(stepPin1, HIGH);**

**delayMicroseconds(motor\_upocasnitev);**

**digitalWrite(stepPin1, LOW);**

**delayMicroseconds(motor\_upocasnitev);**

**}**

**}**

**while (1) //Po končanem skenu se prograam ustavi**

**{**

**/\* code \*/**

**}**

**}**

# Komentar

Vse kar sem hotel ni bilo povsem uresničljivo v podanem časovnem okvirju. Če bi bolj pazljivo nosil izdelek in dal delati PCB ploščo v tovarno bi lahko prihranil na času in problemi bi se manj krat pojavili. Lahko bi tudi takoj po končanem vezju naredil ohišje zanj saj bi si s tem olajšal prenašanje in stabilnost vezja.