

Šolski center Ljubljana, Srednja strojna in kemijska šola Ljubljana



INTERAKTIVNA 3D OČALA ZA MERJENJE

Raziskovalna naloga s področja mehatronike

Avtorji: Rok SKVARČA, Ana RIBIČ

Mentor: Robert JAMNIK

in Matija VRHOVEC

Vrhniko, 22.2.2020

KAZALO

| | |
|---|-----------|
| POVZETEK | 3 |
| 1 UVOD | 3 |
| 2 TEORETIČNI DEL | 4 |
| 2.1 Novejša zgodovina pametnih očal..... | 4 |
| 2.2 Zgodovina in nastanek mikrokrmilnika | 4 |
| 2.3 Googlova pametna očala..... | 5 |
| 2.4 Intelova pametna očala Vaunt..... | 6 |
| 3 EKSPERIMENTALNI DEL | 7 |
| 3.1 Koncept celotnega sistema | 7 |
| 3.2 Izdelava mikrokrmilniške plošče..... | 7 |
| 3.2.1 Mikroprocesor ATmega 328p | 7 |
| 3.2.2 Izdelava tiskane ploščice..... | 8 |
| 3.2.3 Izdelava tiskanega vezja | 9 |
| 3.2.4 Sestavljanje vseh elementov | 10 |
| 3.2.5 Priklop in uporaba povezovalnega kabla TTL 323 | 10 |
| 3.2.3 Priklop mikrokrmilnika na računalnik | 11 |
| 3.3 Preostali elementi pametnih očal | 12 |
| 3.3.1 OLED zaslon | 12 |
| 3.3.1.1 Način delovanja..... | 12 |
| 3.3.1.2 Gonilniki in njegovi vmesniki | 13 |
| 3.3.1.3 Zahteve za napajanje | 13 |
| 3.3.1.4 OLED pomnilniški zemljevid..... | 14 |
| 3.3.1.5 Priključitev OLED zaslona na krmilnik | 14 |
| 3.3.2 Zasnova in izdelava ohišja..... | 15 |
| 3.3.3 Potenciometer in očala..... | 16 |
| 3.5 Programiranje..... | 16 |
| 3.5.1 Namestitev knjižnice | 16 |
| 3.5.2 Arduino koda – prikazovanje besedila | 17 |
| 3.5.3 Arduino koda – prikazovanje besedila (ime šole)..... | 17 |
| 3.5.4 Arduino koda – drsenje besedila | 18 |
| 3.5.5 Arduino koda – prikaz slike (Logotip šole) | 18 |
| 3.5.6 Arduino koda - potenciometer | 19 |
| 4 REZULTATI IN RAZPRAVA | 20 |
| 5 ZAKLJUČEK | 20 |
| 6 LITERATURA | 21 |

POVZETEK

V današnjem času je vedno bolj pomembna varnost pri delu. Varnost pri delu je bila naš motiv oziroma povod za to raziskovalno nalogo, ki smo jo izvedli. Osnovna ideja raziskave je, da bi izdelali merilno napravo, ki bo omogočala varno merjenje napetosti, toka itd. brez bojazni, da bi naredili kakšen kratek stik ali prijeli kakšno žico, ki je pod napetostjo medtem, ko bi pogled odvrčali stran od mesta merjenja zato, da bi videli odčitek, kot pri tradicionalnih napravah za merjenje. Seveda je to teorijo ali hipotezo potrebno raziskati ali je sploh smiselna.

Zasnovati smo morali prenos informacij iz mesta merjenja do tega, da bi merilec izmerjeno vrednost videl na očalih. Problematično je bilo predvsem to, da bi s to merilno napravo omejili merilčev pogled in kako namestiti napravo na očala, ne da bi merilca ovirala, kakšna očala sploh izbrati itd.?

Rezultati so pokazali, da se ta merilna naprava zelo dobro obnese in močno izboljša varnost pri delu. Seveda je še veliko prostora za izboljšave te naprave.

Ključne besede: varnost, očala, prenos informacij, merilec, naprava.

1 UVOD

Raziskovalna naloga predstavlja in opisuje, kako izboljšati varnost pri delu ali pa nam olajša delo. Tako smo izhajali iz hipoteze, da nam lahko nova tehnologija omogoči večjo varnost pri delu. Namen naloge je bil izboljšati varnost pri merjenju, v našem primeru, napetosti. Zato smo izdelali prototipna 3D interaktivna merilna očala.

Za nas je bil to še poseben izziv, ker to področje še ni množično raziskano. Menimo, da imajo pametna očala velik potencial, da se bodo v prihodnosti še razvijala in postala uporabna za navadne smrtnike, kot npr. pametni telefoni.



Slika1: Intelova pametna očala.

izpostavili, da očala ne izgledajo kot tehnološka igračka. Na zunaj morajo biti enaka ostalim očalom. Svoja očala so poimenovali **Vaunt**.

Zagotovo ste že videli Googlova očala oz. naglavni pripomoček, ki pred očesom ponuja manjši zaslon, na katerem se izpisujejo obvestila. Ker se je izkazalo, da ljudi ne zanima masovna uporaba tega pripomočka, se je Google osredotočil na tovarne in podjetja, ki jim uporaba pametnih očal dejansko pomaga. Intel pa je na drugi strani razvil očala, ki bi lahko čez nekaj let stala pred očmi vsakega uporabnika.

Pri Intelu so kot ključno lastnost

2 TEORETIČNI DEL

V teoretičnem delu podrobno predstavimo zgodovino nekaterih komponent, razvoj pametnih očal pri drugih proizvajalcih in tehnologijo, ki so jo vgradili.

2.1 Novejša zgodovina pametnih očal

Pametna očala so v osnovi običajna očala z dodano tehnologijo. Pionir na tem področju je bil Steve Mann, ki je med prvimi obsežneje delal na tem področju. Njegova očala imenovana »EyeTap« so se nosila pred očmi in uporabljala sistem cepilnik žarkov, ki je prikazovalo sliko prenesel na oba očesa. Očala so bila povezana s mini računalnikom, ki se ga je nosilo v žepu. Očala je predstavil leta 1999. Pred tem pa so pametna očala izgledala kot nekakšna čelada. Imela so več negativnih kot pozitivnih lastnosti, saj so bila velika, moteča in nerodna. Zaradi nepraktičnosti se niso uveljavila, saj v tistem času še ni bilo tako napredne tehnologije, da bi se razvijala še naprej. Do preboja je prišlo leta 2012, ko je podjetje Google predstavil svoja pametna očala. Ta očala so dala povod za razvijanje in izboljševanje pametnih očal.



Slika 2: Steve Mann s svojimi pametnimi očali

2.2 Zgodovina in nastanek mikrokrmilnika

Je majhen računalnik na kovinskem – oksidnem – polprevodniku (ang. MOS). Vsebuje enega ali več procesorjev, notranji pomnilnik ter vmesnike. V nasprotju z mikroprocesorji, ki se uporabljajo v osebni računalnikih, se mikrokontrolerji uporabljajo v samodejno nadzorovanih izdelkih in napravah, kot so pisarniški stroji, električno orodje, daljinski upravljalniki.

Inženirja Gary Boone in Michael Cochran sta zaslužna za uspešno ustvarjanje prvega mikrokontrolerja leta 1971. Rezultat njunega dela je bil TMS 1000. TMS 1000 je družina mikrokrmilnikov, ki so bili prvi popolni računalniki na čipu. Namenjeni so bili vgrajenim sistemom v avtomobilih, igrah in merilnih inštrumentih. Bili so tudi prvi komercialni mikrokrmilniki velike količine. Leta 1974 je bilo mogoče čipe v tej družini kupiti samo za 2 dolarja. Delno kot odgovor na obstoj TMS 1000, je Intel razvil računalniški sistem na čipu Intel 8048. Predstavljen je bil leta 1976 in je bil prvi Intelov mikrokontroler. Uporabljen je bil kot

procesor IBM-ove tipkovnice. 8051 mikrokontroler je bil uveden leta 1980 in je eden izmed najbolj priljubljenih mikrokontrolerjev. Uporablja se celo zdaj in velja za enega najbolj dolgoživih. V devetdesetih letih prejšnjega stoletja so na trgu elektronike začeli preplavljati napredni mikrokontrolerji z električno izbrisljivimi in programirljivimi pomnilniki ROM, kot je bliskovni pomnilnik. Edinstvena značilnost teh mikrokontrolerjev je, da jih je mogoče programirati, izbrisati in reprogramirati s pomočjo samo električnih signalov. Tehnologijo bliskovnega pomnilnika uporablja veliko trenutno uporabljanih mikrokontrolerjev, kot sta na primer Atmel in Microchip. Današnji mikrokontrolerji so postali manjši in vitkejši, hkrati pa še močnejši. Na primer, na voljo je tako drobnih mikrokontrolerjev, dovolj majhnih in poceni, da jih lahko uporabljamo v preprostih izdelkih, kot so zobne ščetke in igrače.



Slika 3: TMS 1000, procesor iz leta 1980



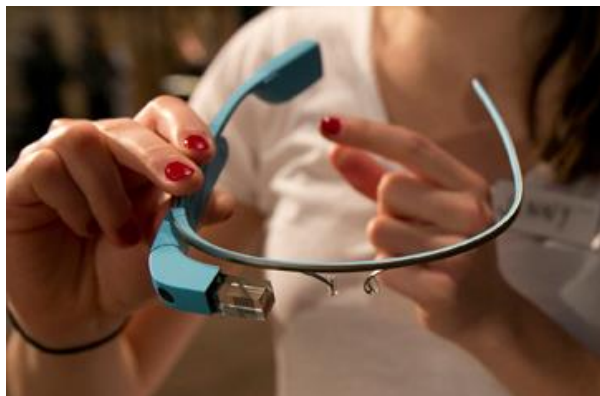
Slika 4: Intel 8048

2.3 Googlova pametna očala

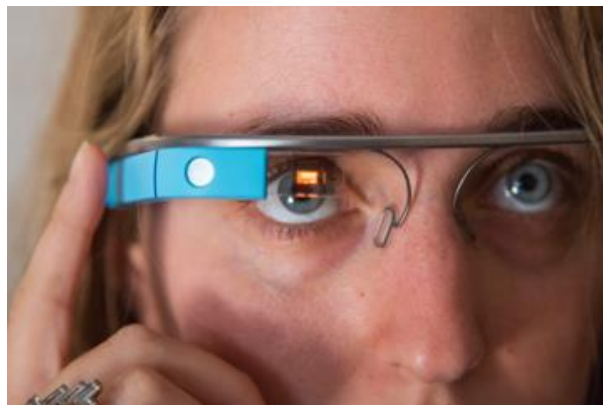
Google glass ali preprosto Glass je blagovna znamka pametnih očal, oblikovan v obliki dvojnih očal. Prototip Google glass je spominjal na standardna očala, ki so nadomestila zaslon za glavo (ang. HUD). Sredi leta 2011 so izdelali prototip, ki je tehtal 3,6 kilogramov, do leta 2013 so bila lažja od povprečnega para sončnih očal. Izdelek je bil javno objavljen aprila 2012, kjer so pokazali kako se Google Glass lahko uporablja za snemanje videov. Javno so bila dostopna šele leta 2014. Do današnjega časa so jih razvijali in predstavili kar nekaj različic. Uporabljajo se pri v medicini, kirurških posegih in pomagajo otrokom, ki imajo avtizem.

Očala lahko nadzorujemo z uporabo kapacitivne sledilne ploščice vzdolž desne strani očal, ali pa z glasovnimi ukazi. Od leta 2014 so v očalih mikroprocesor na osnovi ARM, video procesorji in pomnilniški vmesnik. Po specifikacijah je čip sposoben predvajati video z ločljivostjo 1080p in 30 sličic na sekundo. Na glavni plošči je nameščen tudi pogon SanDisk za pomnilnik – 16 GB, čeprav je uporabniku na voljo le 12 GB. Na očalih so še nameščeni senzorji bližine, zunanje svetlobe in inercialni senzor, ki zaznava gibanje. Očala omogočajo tudi fotografiranje v ločljivost 5 milijonov slikovnih pik ter zajemanje videov v ločljivosti 720p. Najbolj zanimiva stvar na teh očalih pa je zvočnik kostne prevodnosti. To pomeni, da zvočnik pošilja vibracije, ki potujejo skozi lobanjo, do vašega notranjega ušesa – ni potrebno nositi slušalk. Skupna uporaba kamere in zvočnika vam omogočata klicanje z videokonference. Za prenos slike v naše oko pa skrbi prizma. Vse te komponente potrebujejo energijo za delovanje. To energijo

zagotavlja baterija z zmogljivostjo 2,1 vatne ure. Google navaja, da polnjenje baterije traja le 45 minut.



Slika 5: Prikazuje prizmo googlovih pametnih očal



Slika 6: Prototip google glasses "Eksplorer"

2.4 Intelova pametna očala Vaunt

Ker se je izkazalo, da ljudi ne zanima masovna uporaba Googlovih pametnih očal, se je Google pred lani osredotočil na tovarne in podjetja, ki jim uporaba pametnih očal dejansko pomaga. Intel pa na drugi strani razvija očala, ki bi lahko čez nekaj let stala pred očmi dosti večje ciljne publike, poimenovali so jih Vaunt. V osnovi gre za laser, ki prek majhnega ogledala projicira tekst ali ikone neposredno na mrežnico očesa, zato tu ni nobenega zaslona, na katerega bi se moral osredotočiti. Morebitne skrbi, ker očala v oči projicirajo sliko z laserjem, pa pri Intelu odpravljajo z zagotovilom, da je laser tako šibek, da je povsem varen. Očala bodo poleg laserja vsebovala še procesor za delovanje aplikacij, kompas in merilnik pospeškov, z mobilnikom pa jih bo mogoče povezati prek povezave bluetooth. Končne različice pa bi lahko vključevale tudi prepoznavanje glasu. Ta očala naj bi bila posebna predvsem po tem, da naj bi zgledala kot običajna očala. Uporabili bi jih lahko med kuhanjem, kjer bi se nam prikazovala navodila po receptu, ali pa jih lahko uporabimo kot navigacijo in še veliko več. Čeprav nekateri viri poročajo da očala že delujejo, so načeloma še v zelo zgodnji razvojni fazi.

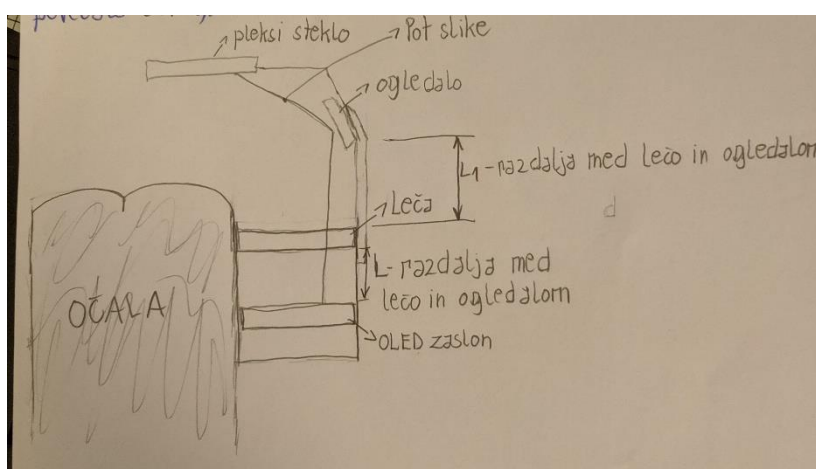


Slika 7: Intelova pametna očala Vaunt

3 EKSPERIMENTALNI DEL

3.1 Koncept celotnega sistema

Za začetek vam bomo opisali koncept naših pametnih očalih. Izbrali smo si navadna varnostna očala, na katerih bo nameščeno ohišje. V ohišju bo OLED zaslon in po potrebi še leča. Zaslon nam bo služil kot projektor, ki nam bo projiciral izpisano besedilo oz. sliko na ogledalo. Ogledalo in akrilno steklo bosta nameščena na nosilcu, ki bo pritrjen na ohišju. Od ogledala se bo slika odbila na akrilno steklo. Akrilno steklo nam bo služilo kot projekcorno »platno«, v katerem bomo videli meritev. S tem odbojem slike od ogledala v akrilno steklo smo dosegli, da se slika prav obrne, sicer če bi gledali direktno v ogledalo bi videli obrnjeno sliko. OLED zaslon v ohišju bo povezan z mikrokrmilnikom, ki se ga bo nosilo v žepu ali pa kjer koli drugje. Do mikrokrmilnika pa bosta speljani še dve žici s katerima bomo opravljali meritev.

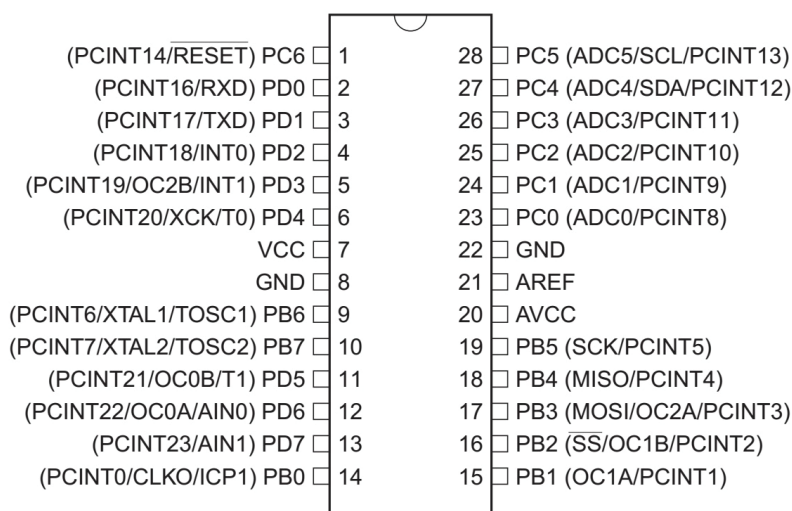


Slika 8: Začetki načrtovanja

3.2 Izdelava mikrokrmilniške plošče

3.2.1 Mikroprocesor ATmega 328p

ATmega328 je mikrokontroler, ki ga je ustvarilo podjetje Atmel in sicer v družini megaAVR. Je 8-bitni mikrokontroler in deluje med 1,8-5,5 voltov. Ima 28 priključnih nogic oz. pinov, največja delovna frekvenca pa je 20 MHz. Da bo mikrokrmilnik prav deloval, si moramo izbrati takega, ki ima že naložen zagonski nalagalnik (ang. boot loader). Ta nalagalnik lahko zažene tudi več manjših aplikacij, ki se lahko zaganjajo zaporedoma ter tako naložijo in zaženejo vse potrebne module. Od leta 2013 se ATmega328 pogosto uporablja v mnogih projektih in avtonomnih sistemih, kjer je potreben preprost in poceni mikrokontroler. Morda je najpogostejša izvedba tega čipa na priljubljeni platformi Arduino, in sicer modelov Arduino Uno in Arduino Nano.



Slika 9: Razpored priključkov na procesorju ATmega328p

| Št. | Funkcija | opomba |
|------------|---------------|---------------------------|
| 1 | reset | ponovni zagon |
| 2 | RXD | sprejem podatkov iz PC-ja |
| 3 | TXD | pošiljanje podatkov v PC |
| 7,20 | VCC | + pol napajanja |
| 8,22 | GND | - pol napajanja |
| 9,10 | XTAL | priklop kristala |
| 23-28 | analogni pin | analogni vhod |
| 2-6, 11-19 | digitalni pin | digitalni priključek |

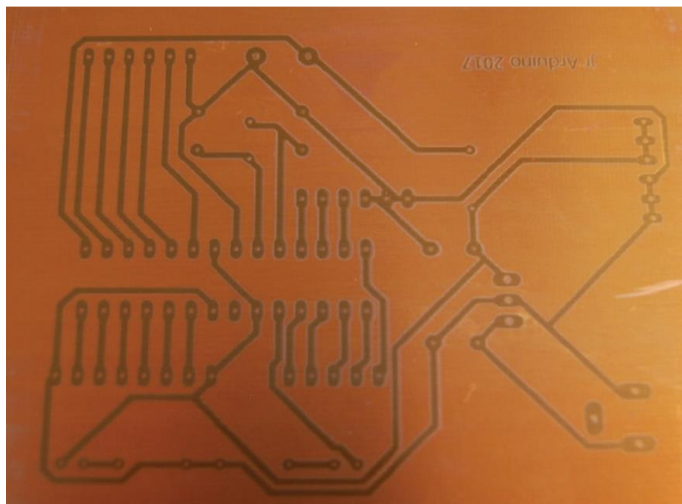
Tabela 1: Funkcije priključkov na procesorju

3.2.2 Izdelava tiskane ploščice

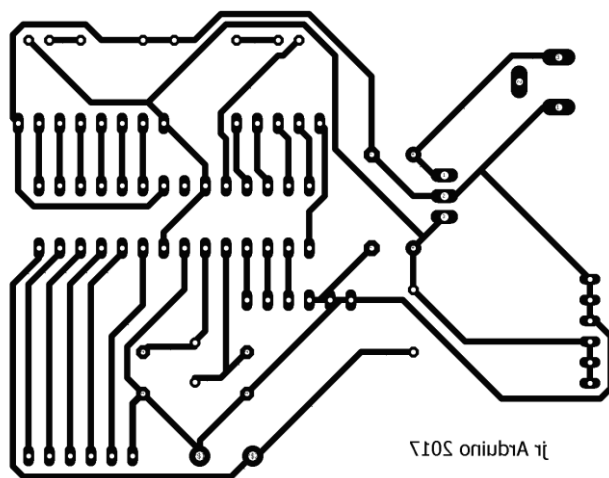
Shema je povzeta po tovarniških načrtih za mikroprocesor ATmega 328P. Tiskano vezje je narejeno tako, da je poleg mikroprocesorja dovolj prostora še za priklop dodatnih modulov oziroma povezovalnih žic. Zato je na ploščici tudi nekaj več priključnih letev, dodali smo pa še napajanje, ki je izvedeno s klasičnim napetostnim regulatorjem 7805. Tako bomo mikrokrmilnik lahko priključili na zunanje napajanje od 6 do 15 V. Pri uporabi zunanjega napajanja moramo paziti na pravilen priklop oziroma pravilno polariteto konektorja. V našem mikrokrmilniku je konektor za napajanje priklopljen tako, da je srednji priključek negativni pol zunanji pa pozitivni. Nekaj zaščite pri napačnem priklopu nudi tudi napetostni regulator, ki ima vgrajeno zaščito proti napačnemu priklopu. V tabeli 1 so na kratko opisani posamezni priključki na mikroprocesorju. Priključki, ki služijo za napajanje, reset in priklop kristala imajo običajno samo eno funkcijo, ostali pa imajo lahko tudi več funkcij. S tem mislimo na različne vrste vhodnih signalov, ki so lahko analogni ali digitalni. Enako velja za izhodne signale. Zato je najprimerneje, če si pri projektih pomagamo s shemo, na kateri so posamezni priključki označeni in opisani. Seveda je pomembno tudi, katere priključke uporabimo oziroma aktiviramo v programu.

3.2.3 Izdelava tiskanega vezja

S pomočjo tiskalnika smo naredili film za tiskano vezje. Na prosojnico smo v dveh izvodih natisnili tiskano vezje. Dve natisnjeni sliki smo združili, da so črte, ki predstavljajo vezje, res neprosojne za ultravijolično svetlobo. Nato smo fotoploščico osvetlili skozi film in jo razvili z razvijalcem. Čas osvetlitve je odvisen od svetlobnega vira in je lahko od nekaj deset sekund do nekaj minut. Za osvetljevanje smo uporabil štiri UV-cevi dolžine 400 nm. Razvijalec pripravimo tako, da naredimo raztopino 7 g NaOH v litru vode. Razvijanje traja do nekaj minut. Ko je fotolak razvit ga speremo z vodo. Sledi jedkanje bakra. Najvarnejša je uporaba raztopine železovega triklorida (FeCl_3). Res je, da postopek traja nekoliko dlje in po njem ostanejo precej zoprni madeži, vendar je varnejši, kot če bi delal s klorovodikovo kislino in vodikovim peroksidom. Izdelava tiskanih vezij doma velja danes bolj za hobi kot resno delo, vendar je na koncu ob uspešnem izdelku zadovoljstvo mnogo večje. Načrt in tiskano vezje sta narejena s programom Eagle. Kot vidimo, je za izdelavo vezja potrebnih zelo malo sestavnih elementov. Uporabili smo res samo tiste, ki so nujni za osnovno delovanje. Tako imamo poleg mikroprocesorja Atmega328P še 16-MHz kristal, dva kondenzatorja po 22 pF, dva kondenzatorja 10 μF , upor 10



Slika 10: Fotoploščica po razvijanju



Slika 11: Tiskano vezje

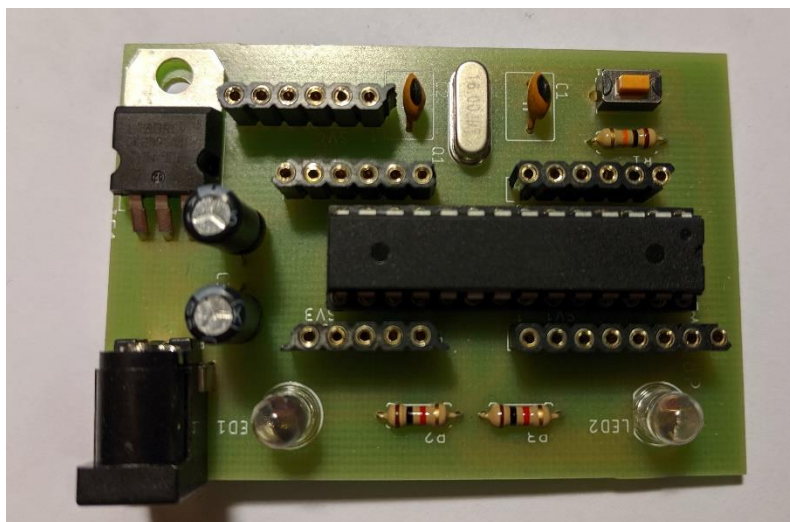
k Ω za

RESET, tipko za RESET (TIPKA MINI 6* 3,5 V =

4,3 mm) in napetostni regulator 7805. Na vezje smo namestili še dve LED diodi, prva je za signalizacijo napajanja, druga pa je priključena na izhod 13. Vse vhodne in izhodne priključke smo povezali na ženske priključne letvice. Tako je mogoče s povezovalnimi žičkami enostavno povezati vhodne in izhodne priključke mikroprocesorja.

3.2.4 Sestavljanje vseh elementov

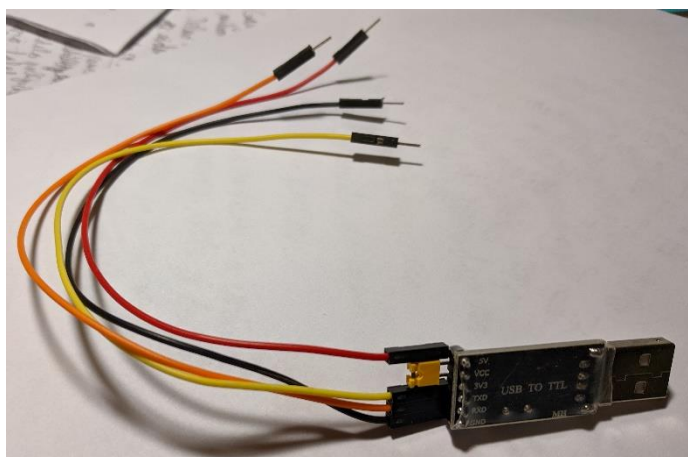
Najprej vstavimo podnožje za mikroprocesor, potem razne konektorje, upore, kondenzatorje, diode, kristale in na koncu še napetostni regulator. Za vgrajevanje smo uporabili spajkalnik in pa dodajni material cin. Spajkalnik smo segreli na približno 300 stopinj. Tak vrstni red je zaželen predvsem zato, da kakšnega elementa po nepotrebem ne uničimo s predolgim segrevanjem. Mikroprocesor obvezno vgradimo na kakovostno podnožje.



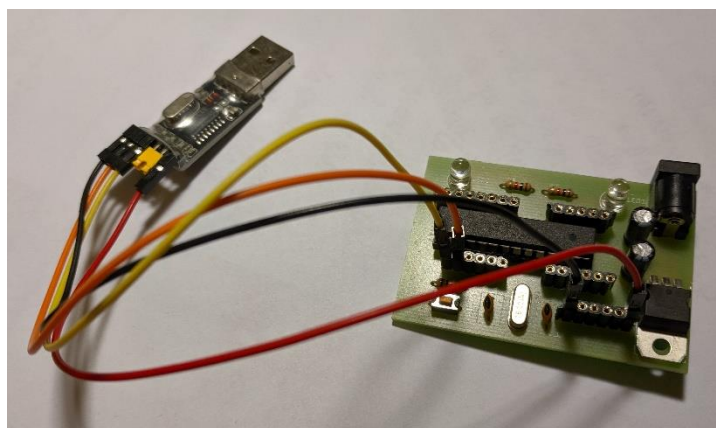
Slika 12: Sestavljen mikrokrmilnik

3.2.5 Priklop in uporaba povezovalnega kabla TTL 323

TTL kabel vsebuje majhno elektronsko vezje, ki je vgrajeno v USB konektor. Vezje poskrbi za ustrezen prenos signalov in protokolov med našim mikroprocesorjem in osebnim računalnikom. Za priklop uporabimo štiri povezovalne žičke, dve za napajanje (rdeča in črna) ter dve za prenos podatkov (oranžna in rumena). Kabel priključimo na mikrokrmilnik tako, da uporabimo rdečo in črno žico za napajanje mikrokrmilnika. Ker kabel omogoča komunikacijo med mikrokrmilnikom in računalnikom, je zato treba priključiti še oranžno in rumeno žičko. Na tiskanem vezju je v ta namen vgrajena ženska priključna letvica, označena s SV2. Uporabimo prvi priključek, ki je povezan na nožico vhoda RXD (nožica 2, glej tabelo) in drugi priključek, ki je povezan na RXT (nožica 3, glej tabelo). Sedaj moramo biti pozorni, da pravilno priključimo kabelski vmesnik, saj so oznake na kablu ravno obratne. To pomeni, da oznako na kablu TXD (oranžna, T – transmit, oddajanje) povežemo na mikroprocesor, ki ima oznako vhoda RXD (R – receive, sprejem), kar je po svoje logično, saj kabelski vmesnik oddaja, mikroprocesor pa sprejema. Enako velja za drugi priklop, kjer je oznaka na kablu RXD (rumena, R – receive, sprejemanje), na mikroprocesorju pa je to vhodna nožica 3, ki ima oznako TXD (T – transmit, oddajanje).



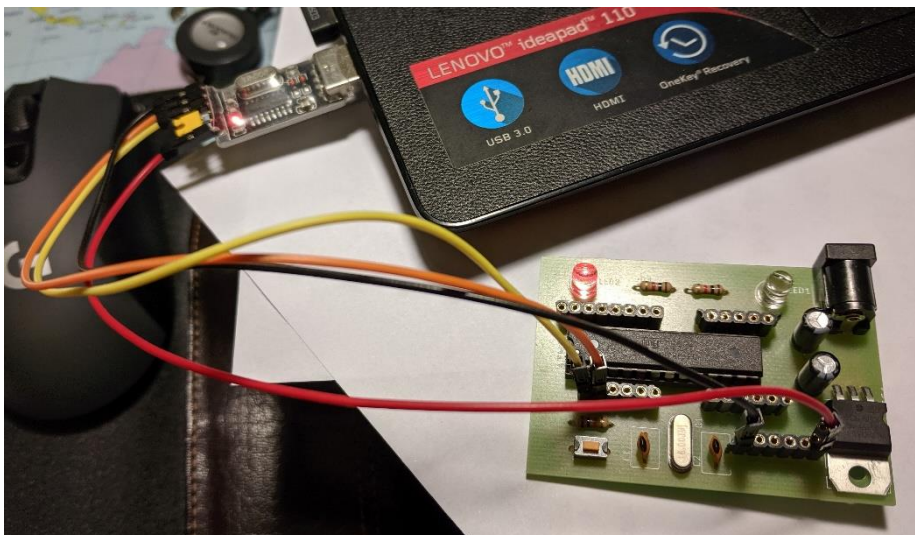
Slika 13: TTL kabel



Slika 14: TTL kabel povezan s mikrokrmilnikom

3.2.3 Priklop mikrokrmilnika na računalnik

Za napajanje uporabimo samo TTL kabel. TTL kabel povežemo z našim mikrokrmilnikom, in sicer tako, kot je napisano v prejšnjem odstavku. Zaženemo program Arduino in preverimo, če je program zaznal priklopljeno ploščo. Ker je plošča lastne izdelave in priklopljena prek kabla, tokrat ne bomo dobili izpisano, kaj smo priklopili, ampak bodo označena vrata (COM), na katera smo priklopili kabel in mikrokrmilnik. Zdaj moramo paziti, da bomo mikrokrmilnik pripravili do tega, da bo začel sprejemati program. To naredimo tako, da pritisnemo tipko RESET točno takrat, ko se v programskem oknu Arduino pojavi napis, da skica uporablja toliko in toliko bajtov. Pravi trenutek je pomemben zato, ker program Arduino najprej preveri program, šele potem ga začne pošiljati podatkovnemu kablu in s tem tudi našemu mikrokrmilniku. Ko pritisnemo tipko RESET, se mikrokrmilnik ustavi in ponovno zažene ter takrat na svojih vseh tudi preveri, če lahko sprejme kakšen nov program. Če bomo zamudili, ne bo nobene škode, samo postopek bomo morali ponoviti.



Slika 15: Mikrokrmilnik povezan s računalnikom

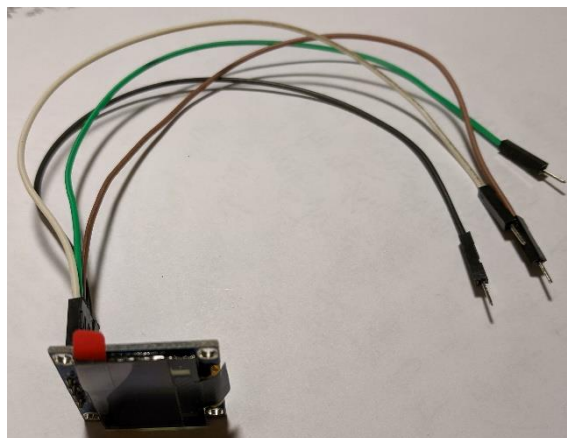
3.3 Preostali elementi pametnih očal

Zdaj ko imamo »možgane« naše naprave, pa potrebujemo še preostale elemente naših očal. Opisani so v nadaljevanju.

3.3.1 OLED zaslon

3.3.1.1 Način delovanja

Organska svetleča dioda (OLED), je svetlobna dioda (LED), v katerem je oddajni sloj film organska spojina. Ta sloj oddaja svetlobo kot odgovor na električni tok. Organski sloj je med dvema elektrodama; običajno je vsaj ena od teh elektrod prozorna. OLED-ji se uporabljajo za ustvarjanje digitalnih zaslonov v napravah, kot so televizijski zasloni, računalniški monitorji, pametni telefoni, ... Zaslون OLED deluje brez osvetlitve ozadja, ker oddaja vidno svetlobo. Tako lahko prikaže globoko črno raven in je lahko tanjši in lažji od zaslona s tekočimi kristali (LCD). V pogojih z nizko osvetlitvijo (na primer v temni sobi) lahko OLED zaslon doseže večje kontrastno razmerje kot LCD. Izbrali smo OLED zaslon SSD1306. Za ta zaslon smo se odločili, ker je lažji in ustvarja svetlejšo in bolj jasno sliko kot LCD zaslon.



Slika 16: Naš OLED zaslon

3.3.1.2 Gonilniki in njegovi vmesniki

V središču modula je močan gonilnik CMOS OLED z enim čipom – SSD1306. Z mikrokontrolerjem lahko komunicira na več načinov, vključno z I2C in SPI. I2C je protokol za serijsko komunikacijo, zato se podatki prenašajo po kosih vzdolž ene žice. SPI je skupen komunikacijski protokol, ki ga uporabljajo številne različne naprave. Ena edinstvena prednost SPI je dejstvo, da se podatki lahko prenašajo brez prekinitev. Vsako število bitov lahko pošljete ali prejmete v neprekinjenem toku. Z I2C se podatki pošiljajo v paketih, omejenih na določeno število bitov. SPI je splošno hitrejši od I2C, vendar zahteva več vhodnih in izhodnih zatičev. Medtem ko I2C potrebuje samo dva zatiča in jih lahko delite z drugimi I2C napravami. Gre za kompromis med zatiči in hitrostjo. Mi smo si izbrali OLED z I2C komunikacijo, ker nimamo na voljo toliko zatičev in ker nam hitrost ni v prvem planu. Zahvaljujoč vsestranskosti krmilnika SSD1306, je zaslon na voljo v različnih velikostih in barvah. Na primer 128*64, 128*32, z belimi OLED-ji, modrimi OLED-ji in dvojnimi barvami OLED-ji. Dobra novica je tudi to, da so vsi ti prikazovalniki zamenljivi.



Slika 17: Ležeči OLED zaslon (z I2C komunikacijo)



Slika 18: OLED zaslon s SPI komunikacijo

3.3.1.3 Zahteve za napajanje

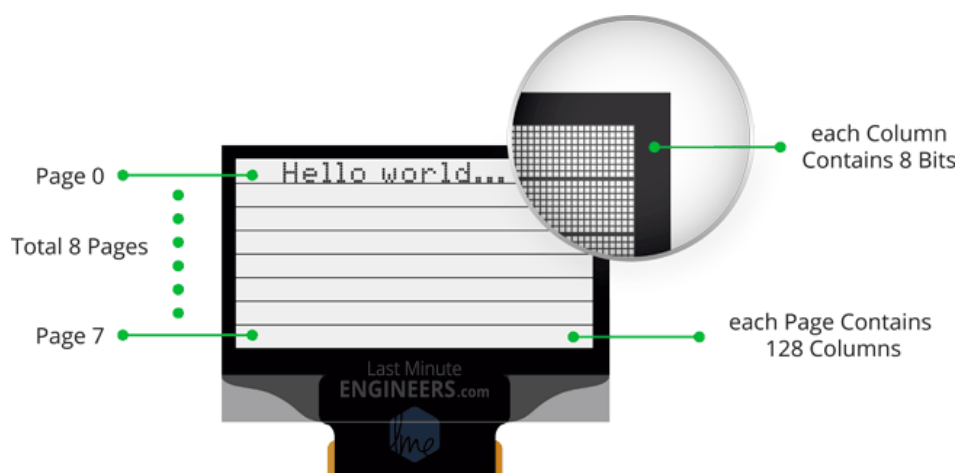
Zaslon OLED deluje brez osvetlitve ozadja, ker daje svojo luč. Zato ima zaslon tako visok kontrast, izjemno širok kot gledanja in lahko prikazuje globoke črne ravni. Odsotnost osvetlitve ozadja znatno zmanjša moč, potrebno za zagon OLED-a. V povprečju zaslon porabi približno 20 mA toka, čeprav je odvisno od tega, koliko zaslona je osvetljeno. Delovna napetost regulatorja SSD1306 je od 1,65 V do 3,3 V, medtem ko OLED plošča potrebuje napajalno napetost od 7 do 15 V. Vse te različne zahteve glede moči so zadostne z uporabo notranjega vezja črpalke za polnjenje. To omogoča enostavno povezavo z Arduino ali katerim koli 5V logičnim mikrokontrolerjem brez uporabe nobenega pretvornika ravni logike.

3.3.1.4 OLED pomnilniški zemljevid

Ne glede na velikost modula OLED ima gonilnik SSD1306 vgrajen 1KB grafični prikaz podatkov RAM (GDDRAM) za zaslon, ki vsebuje bitni vzorec za prikaz. To območje 1K pomnilnika je organizirano na 8 straneh (od 0 do 7). Vsaka stran vsebuje 128 stolpcev / segmentov (blok od 0 do 127). In vsak stolpec lahko shrani 8 bitov podatkov (od 0 do 7). To zagotovo pove, da imamo:

$8 \text{ strani} \times 128 \text{ segmentov} \times 8 \text{ bitov podatkov} = 8192 \text{ bitov} = 1024 \text{ bajtov} = 1 \text{ KB pomnilnika}$

Vsak bit predstavlja določen OLED piksel na zaslonu, ki ga je mogoče programsko vklopiti ali izklopiti.



Slika 19: Celoten 1K pomnilnik s stranmi, segmenti in podatki

3.3.1.5 Priključitev OLED zaslona na krmilnik

Povezave so dokaj preproste. Začeli smo tako, da smo priključili VCC pin na 5V izhod mikrokrmilnika in GND priključili na GND krmilnika. Zdaj ostaneta še samo zatiča, ki se uporabljata za komunikacijo I2C. SCL pin smo povezali na digitalni pin (21) na procesorju ter (SDA) pin na pin (20) na procesorju. SDA je podatkovna linija, SCL pa urna linija.



Slika 20: Razporeditev priključkov na zaslonu

3.3.2 Zasnova in izdelava ohišja

V inventarju smo najprej začeli z risanjem osnove oblike podnožja za ohišje oled zaslona in nato naredili celotno ohišje s pomočjo funkcij extrude in hole. Naredili smo še notranjost za stabilizacijo oled zaslona s funkcijo extrude. Na to ohišje smo dodali še držalo za ogledalo, s katerim bomo kasneje lahko nastavljali jasnost slike in oddaljenost projiciranih podatkov na prosojno folijo.

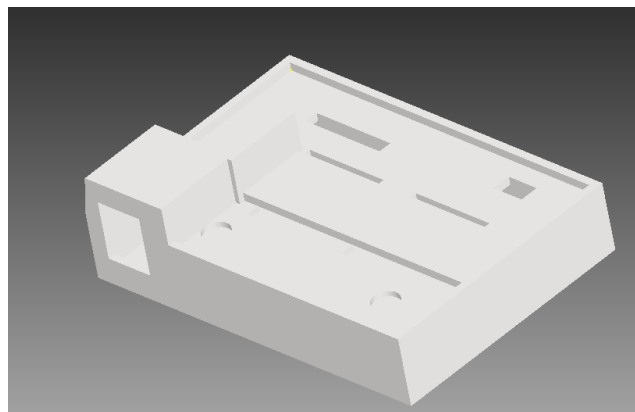
Pokrov smo naredili precej osnoven, ampak uporaben za to ohišje. Omogoča lahek dostop in če je potrebno, menjavo notranjih komponent brez večjih problemov. Za cel postopek smo uporabili samo funkcijo extrude.

Držalo prosojne folije smo zaradi praktičnih razlogov naredili tako, da se lahko pritrdi kar direktno na ogledalo. Za točnost kotov bomo morali izvesti še precej preizkusov in manjših sprememb dokler ne bo popolno. Poskušamo izboljšati kar se le da, v omejenem času in brez večjih sprememb v obliki če ni potrebno.

Poleg ohišja, držala in pokrova smo naredili še kompaktno škatlico za krmilnik in njegovo vezje s katerim bomo vse vodili. Škatlica je narejena tako, da imamo lahek dostop do vseh potrebnih konektorjev, led diod in gumba za resetiranje. Zaradi različnih višin komponent na plošči, je pokrov na nekaterih predelih višji kot na drugih.



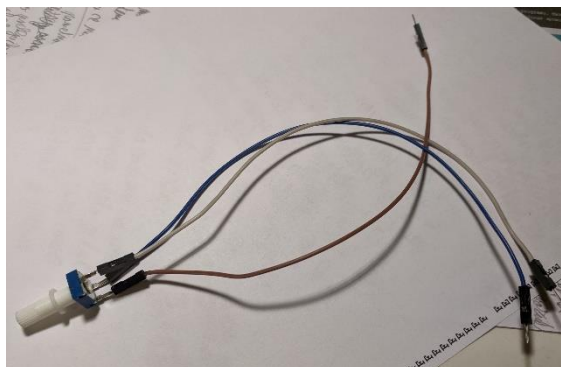
Slika 21: 3D natiskano ohišje pametnih očal



Slika 22: 3D natiskano ohišje mikrokrmilnika

3.3.3 Potenciometer in očala

Potenciometer smo uporabili samo za simuliranje meritve električnega toka. Ko potenciometer privijemo, se na zaslonu spremeni vrednost, kot da bi merili napetost. Za očala smo si izbrali običajna varnostna očala iz akrilnega stekla.



Slika 23: Potenciometer



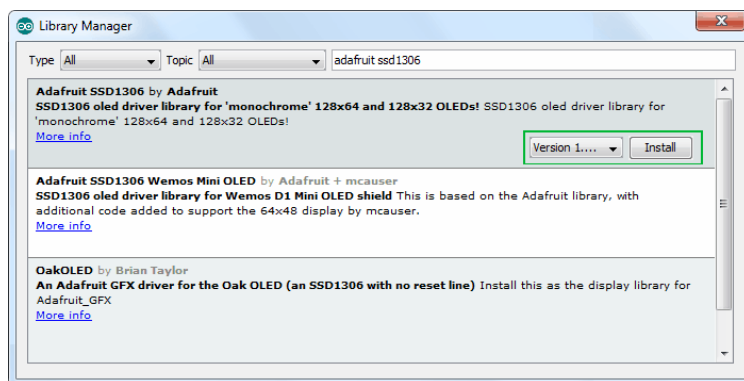
Slika 24: Očala

3.5 Programiranje

Klub temu da imamo že ves strojni del, naprava še ni uporabna. Zato moramo napisati še programski del. Za programiranje smo uporabili program Arduino. Postopek je opisan v nadaljevanju.

3.5.1 Namestitev knjižnice

Krmilnik SSD1306 na zaslonu OLED ima prilagodljive, a zapletene gonilnike. Za uporabo krmilnika SSD1306 je potrebno veliko znanja o naslovu pomnilnika. Na srečo je bila Adafruit-ova knjižnica SSD1306 napisana tako, da je zakrila zapletenosti krmilnika SSD1306, da bomo lahko izdali preproste ukaze za nadzor zaslona. Če želite namestiti knjižnico, se pomaknite do Skica> Vključi knjižnico> Upravljanje knjižnic ... Počakajte, da upravitelj knjižnice prenese indeks knjižnic in posodobi seznam nameščenih knjižnic. Iskanje filtrirajte tako, da vtipkate 'adafruit ssd1306'. Vpisov mora biti nekaj. Poiščite Adafruit SSD1306 by Adafruit. Kliknite ta vnos in izberite Namestitev. Ta knjižnica Adafruit SSD1306 je knjižnica, specifična za strojno



Slika 25: Knjižnica Adafruit SSD1306

opremo, ki upravlja funkcije nižje stopnje. Za prikaz grafičnih primitivov, kot so točke, črte, krogi, pravokotniki itd., Ga morate povezati z knjižnico Adafruit GFX.

3.5.2 Arduino koda – prikazovanje besedila

Skica se začne z vključitvijo štirih knjižnic. SPI.h, Wire.h, Adafruit_GFX.h in Adafruit_SSD1306.h. Čeprav knjižnica SPI.h ni potrebna za zaslone OLED I2C, jo moramo dodati zaradi sestavljanja našega programa.

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

Nato moramo ustvariti objekt Adafruit_SSD1306.h. Konstruktor Adafruit_SSD1306 sprejme Arduino pin številko, na katero je priključen ponastavitveni zatič zaslona. Ker zaslon OLED, ki ga uporabljamo, nima pin RESET, bomo konstruktorju poslali -1, tako da noben od Arduino nožic ne bo uporabljen kot ponastavitev zaslona.

```
Adafruit_SSD1306 display(-1);
```

V nastavitveni funkciji: moramo seznaniti OLED objekt z begin()funkcijo. Funkcija ima dva parametra. Prvi parameter SSD1306_SWITCHCAPVCC vklopi vezje črpalke notranjega polnjenja, medtem ko drugi parameter zagotavlja I2C naslov zaslona OLED. I2C naslov takega OLED prikazovalnega modula je na splošno 0x3C. Fiksno je in ga ni mogoče spremeniti. Nato očistimo medpomnilnik pred tiskanjem prvega sporočila na zaslon.

```
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
display.clearDisplay();
```

3.5.3 Arduino koda – prikazovanje besedila (ime šole)

Za prikaz besedila na zaslonu moramo nastaviti velikost pisave. To lahko storite s klicanjem setTextSize()in posredovanjem velikosti pisave (od 1) kot parametra.

Nato moramo s klicno funkcijo nastaviti barvo pisave setTextColor(). Parameter prehoda BELA za temno ozadje in črno črno za svetlo ozadje. Pred tiskanjem sporočila moramo nastaviti položaj kazalca s klicanjem funkcije setCursor (X, Y). Piksle na zaslonu se nanašajo po njihovih vodoravnih (X) in navpičnih (Y) koordinatah. Koordinatni sistem postavi izvor (0,0) v zgornji levi kot, pozitiven X narašča v desno, pozitiven Y pa navzdol.

Za tiskanje sporočila na zaslon lahko uporabimo preprosto print(" ")ali println(" ")funkcijo, tako kot tiskamo podatke na serijski monitor. Ne pozabite, da bo println () premaknil kazalec v novo vrstico.

Da bi knjižnica lahko izvajala izjemno hitre matematične operacije na medpomnilniku zaslona (več kot 100 sličic na sekundo), klici funkcij za tiskanje vsebine vmesnega blažilnika ne

prenesejo takoj na krmilnik SSD1306. Potreben je `display()` ukaz, ki knjižnici naroči, da opravi prenos v velikem obsegu iz medpomnilnika zaslona v ATmega328P v notranji pomnilnik krmilnika SSD1306. Takoj ko se pomnilnik prenese, se na zaslonu OLED prikažejo slikovne pike, ki ustrezajo zaslonskemu medpomnilniku.

```
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0,28);
display.println("Srednja strojna in kemijska šola Ljubljana");
display.display();
delay(2000);
```

3.5.4 Arduino koda – drsenje besedila

Včasih ne želimo pomikati celotnega zaslona. To lahko storite tako, da posredujete pravilno začetno stran in informacije o ustavitvi strani funkcijam drsenja. Za razlago strani glejte razdelek OLED Memory Map. Ker je na prikazovalniku osem strani od 0 do 7, lahko del zaslona pomikate tako, da kot parametre vnesete določene številke strani.

V našem primeru smo oba parametra prenesli kot `0x00`. To se bo pomaknilo samo po prvi strani (prvih 8 vrstic) zaslona.

```
display.setCursor(0,0);
display.setTextSize(1);
display.println("Srednja strojna");
display.println("in kemijska šola");
display.println("Ljubljana.");
display.display();
display.startscrollright(0x00, 0x00);
```

3.5.5 Arduino koda – prikaz slike (Logotip šole)

Za prikaz bitne slike na zaslonu OLED moramo poklicati `drawBitmap()` funkcijo. Potrebuje šest parametrov. Koordinata zgornjega levega kota, koordinata Y zgornjega levega kota, bajtovski niz enobarvne bitne slike, širina bitne slike v slikovnih pikah, višina bitne slike v slikovnih pikah in Barva. V našem primeru je bitna slika velikosti 128×64 . Koordinate X & Y so nastavljene na 0, širina in višina pa na 128 in 64.

```
display.drawBitmap(0, 0, Logo, 128, 64, WHITE);
display.display();
```

Toda preden lahko pokličemo funkcijo `drawBitmap ()`, potrebujemo sliko za risanje. Ne pozabite, da je ločljivost zaslona OLED zaslona 128×64 slikovnih pik, zato slike, večje od njega, ne bodo prikazane pravilno. Če želite dobiti pravilno velikost slike, lahko uporabite svoje

najljubše programe za risanje, kot so Inkscape, Photoshop, Paint itd. In nastavite velikost platna na 128×64 slikovnih pik. Vzeli smo sliko logotipa naše šole in jo s Paint pretvorili v 128×64 slikovnih pik in jo shranili kot .bmp. Ko imaš bitmap, je čas, da ga pretvoriš v matriko, ki jo lahko razume krmilnik OLED SSD1306. To je mogoče storiti na dva načina: spletna metoda z uporabo image2cpp in metoda brez povezave z uporabo LCD pomočnika. Mi smo uporabili spletno metodo.

3.5.6 Arduino koda - potenciometer

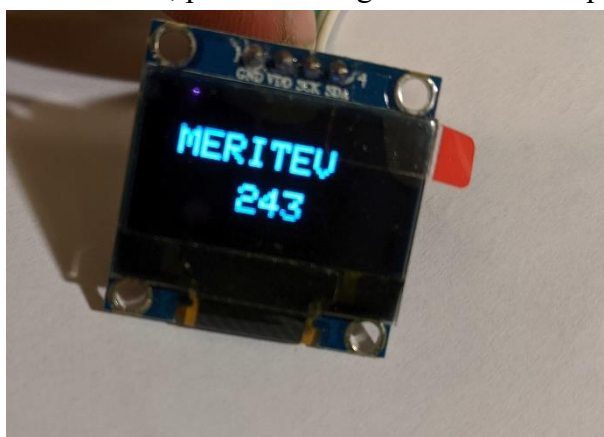
Najprej moramo zapisati začetne pogoje. Vrednost senzorja in merjena napetost je enaka nič. Z funkcijo int. Spremenimo vrednost v številko. S to funkcijo mikrokrmilnik razume, da je to številka. Na koncu še zapišemo ne kateri priključek na mikrokrmilniku smo priključili potenciometer.

```
int sensorValue = 0;
int voltage = 0;
const int analogInPin = PC0;
```

In na koncu napišemo še drugi del programa, ki bo vrednost izpisal na zaslonu.

```
display.clearDisplay();
display.setTextSize(2);
display.setTextColor(WHITE);
display.setCursor(10,15);
display.println("MERITEV");
sensorValue = analogRead(analogInPin);
voltage = map(sensorValue, 0, 1023, 0, 500);
display.setCursor(45,40);
display.print(voltage);
display.display();
```

In tako je naš programski del za očala napisan. Celoten program najprej preverimo, nato pa še naložimo na mikrokrmilnik. In če smo vse naredil prav, bi se moralo na zaslonu izpisati besedilo, besedilo se enkrat zadržalo, prikazal se logo šole in nato se prikazala meritev.



Slika 26: OLED zaslon, ki prikazuje meritev

4 REZULTATI IN RAZPRAVA

Razpravo in rezultate bomo združili, saj bomo tako lažje razložili rezultate na vprašanja, ki so se pojavila.

Zanimalo nas je ali hipoteza, da nova tehnologija omogoča večjo varnost pri delu, drži. To smo poskušali ugotoviti s prototipnimi 3D interaktivnimi merilnimi očali. Ko merilec meri napetost v tokokrogu, pri tradicionalnih napravah za merjenje ostaja nevarnost, da pride do kratkega stika ali prijema kakšne žice, ki je pod napetostjo. S pametnimi merilnimi očali bi zmanjšali nevarnosti, saj bi merilec enostavno napetost odčital na zaslončku, ki je pritrjen na očalih in pri tem ne bi pretirano odvrčal pogleda. Hipotezo smo s preizkušanjem potrdili. S pomočjo vezja, kablov, OLED zaslona, ogledala, zaslona, na katerem se prikaže meritev, in seveda očal, varneje izvedemo meritev. Tako se izognemo morebitnim nevarnostim, ki pretijo, kadar odvrnemo pogled od mesta merjenja.

Zanimalo nas je tudi, ali bo naprava ovirala merilčev pogled, ko bo ta pritrjena na očala. Tudi tu smo ugotovili, da ni razloga skrb, saj smo zaslon fiksirali višje od merilčevega pogleda, torej ga ta ne ovira pri delu.

Postavilo se je tudi vprašanje glede leče, ki smo jo imeli. Bila je namreč malo slabša in motna, zato se številke niso videle oz. dale razbrati na končnem zaslonu.

5 ZAKLJUČEK

Naša pametna očala so izdelana preprosteje kot npr. Googlova ali Intelova, ki so vsebovala zelo komplicirane komponente. Izdela jih lahko vsak v najpreprostejši delavnici, saj so namenjena hobijski mehatroniki. Očala vsebujejo malo komponent, ki se ob okvari lahko enostavno zamenjajo. Tudi finančno so zelo dostopna, saj smo zanje odšteli manj kot 20€. Upamo, da vam je bila predstavitev všeč in da smo vas mogoče navdušili za pametna očala.

6 LITERATURA

- Potočnik, A. (2018). *Minimalistična pametna očala brez kamere*. Pridobljeno 22.februarja 2020 s <https://www.dnevnik.si/1042801218>
- Krajnik, M. (6.2.2018). *Intel razvil pametna očala, ki izgledajo kot navadna očala*. Pridobljeno 22. februarja 2020 s <http://dne.ena.com/Gadgeti/intel-razvil-pametna-ocala-ki-izgledajo-kot-navadna-ocala.html>
- Ravi, (31.8.2018). *Interfacing 128*64 OLED Graphic Display with Arduino – Hook-up Guide and Tutorial*. Pridobljeno 22. februar 2020 s <https://www.electronicshub.org/interfacing-128x64-oled-graphic-display-with-arduino/>
- Interface OLED Graphic Display Module with Arduino*. (b.d) Na [Lastminuteengineers.com](https://lastminuteengineers.com/oled-display-arduino-tutorial/). Pridobljeno 22. februar 2020 s <https://lastminuteengineers.com/oled-display-arduino-tutorial/>
- OLED. (b.d) Na [en.wikipedia.org](https://en.wikipedia.org/wiki/OLED). Pridobljeno 22. februar 2020 s <https://en.wikipedia.org/wiki/OLED>
- Basi of the I2C communication protocol*. (b.d) Na [Circuitbasics.com](https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/). Pridobljeno 22. februar 2020 s <https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/>
- John, (23.10.2013). *Microcontroller – Invention History and Story Behind the Scenes*. Pridobljeno 22. februar 2020 s <http://www.circuitstoday.com/microcontroller-invention-history>
- ATmega328. (b.d) Na [en.wikipedia.org](https://en.wikipedia.org/wiki/ATmega328). Pridobljeno 22. februar 2020 s <https://en.wikipedia.org/wiki/ATmega328>
- Strickland, J. (2014). *How Google Glass Works*. Pridobljeno 22. februar 2020 s <https://electronics.howstuffworks.com/gadgets/other-gadgets/project-glass.htm>
- Jamnik, R (2014). Logični mikrokrmilnik Arduino. *Revija TIM*, 23-31.

