

Prototip samoposlužnog automata temeljen na Arduinu

Zgombić, Bruno

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:539366>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2025-02-07**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET

Prijediplomski sveučilišni studij elektrotehnike

Završni rad

Prototip samoposlužnog automata temeljen na Arduino

Rijeka, kolovoz 2024.

Bruno Zgombić
0069094495

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Prijediplomski sveučilišni studij elektrotehnike

Završni rad

Prototip samoposlužnog automata temeljen na Arduino

Mentor: izv. prof. dr. sc. Jonatan Lerga

Rijeka, kolovoz 2024.

Bruno Zgombić

0069094495

Rijeka, 15.03.2024.

Zavod: Zavod za računarstvo
Predmet: Programska podrška u inženjerstvu

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Bruno Zgombić (0069094495)**
Studij: Sveučilišni prijediplomski studij elektrotehnike (1030)

Zadatak: **Prototip samoposlužnog automata temeljen na Arduinu / Prototype of Self-Service Vending Machine Based on Arduino**

Opis zadatka:

Potrebno je izraditi prototip samoposlužnog automata temeljen na Arudinu. Isti treba imati više odvojenih prostora za svaki proizvod. Sučelje treba sadržavati LCD zaslon, utor za kovanice i tipkovnicu. Po detekciju ubačenih kovanica, korisnik odabire pomoću tipkovnice željeni proizvod kojeg uređaj ubacuje u prostor gdje da ga korisnik može preuzeti.

Rad mora biti napisan prema Uputama za pisanja diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 20.03.2024.

Mentor:
izv. prof. dr. sc. Jonatan Lerga

Predsjednik povjerenstva za
završni ispit:
prof. dr. sc. Dubravko Franković

Izjava

Izjavljujem da sam završni rad pod nazivom „Prototip samoposlužnog automata temeljen na Arduino“ izradio samostalno koristeći stečena znanja i navedenu literaturu.

Rijeka, kolovoz 2024.

Bruno Zgombić

SADRŽAJ

1. UVOD	3
2. AUTOMATIKA I SAMOPOSLUŽNI AUTOMAT	4
3. ELEKTROTEHNIKA	5
3.1. Arduino	5
3.1.1. Arduino Uno R3 SMD	7
3.2. Servo motor	11
3.2.1. Tower Pro MG995 360°	12
3.3. LCD	15
3.3.1. LCD 16x2	15
3.4. Tipkovnica	19
3.4.1. 3x4 Matrix Membrane Keypad	19
3.5. Senzori	21
3.5.1. FC-51 infracrveni senzor blizine	22
3.6. Ostale komponente	23
3.6.1. Spojne žice	24
3.6.2. Eksperimentalna pločica	24
3.6.3. Napajanje	25
4. RAČUNARSTVO	26
4.1. Arduino IDE	26
4.2. Servo motor	28
4.3. LCD zaslon	29
4.4. Tipkovnica	30
4.5. Senzor blizine	31
5. STROJARSTVO	32
5.1. Materijali i strojevi	32

6. IZRADA SAMOPOSLUŽNOG AUTOMATA	34
6.1. Povezivanje električnih elemenata	34
6.2. Programiranje Arduino Uno pločice	36
6.3. Izrada kućišta.....	39
6.4. Princip rada.....	44
7. ZAKLJUČAK.....	46
LITERATURA	47
SAŽETAK I KLJUČNE RIJEČI.....	49
SUMMARY AND KEY WORDS	49
POPIS SLIKA.....	50
POPIS TABLICA.....	51

1. UVOD

Tema ovog završnog rada je prototip samoposlužnog automata (eng. vending machine) temeljen na Arduino. Što je samoposlužni automat? To je svjetski proširen uređaj koji omogućuje jednostavnu i brzu kupovinu svakodnevnih potrepština, najčešće se radi o proizvodima iz prehrambene industrije kao što su sokovi, grickalice, čokolade i drugo. Može se koristiti za različite primjene gdje je potrebna automatizirana prodaja nekog proizvoda. Pod kategoriju samoposlužnih aparata možemo svrstati veliki broj uređaja. Glavni cilj i najvažniji aspekt takvog uređaja je napraviti proces bez prisustva čovjeka na poziciji prodavača određene usluge. Na ovaj način možemo kupiti parkirnu kartu, podignuti novac iz bankomata, kupiti kavu ili grickalice. Sve su to različiti uređaji, ali osnovna svrha je automatizacija procesa te nije potrebna odgovorna osoba za njegovo funkcioniranje.

Povijest samoposlužnog automata je veoma dugačka i seže čak u vrijeme Rimskog Carstva, gdje je, prema istraživanjima prvi put napravljen i korišten u Egipatskoj provinciji od strane grčkog matematičara i inženjera. Takav tip samoposlužnog automata radio je prema isključivo mehaničkim principima pomoću kojih bi po ubacivanju kovanica one pomaknule polugu koja bi izbacila vino. Također postoji primjer korištenja samoposlužnog automata u Engleskoj davne 1615. gdje se po ubacivanju kovanica mogao kupiti duhan. Moderni samoposlužni automati javili su se 1880-tih u Londonu. Bili su vrlo popularni na željezničkim stajalištima zato što su se pomoću njih mogle kupiti razglednice, omotnice ili novine. Ubrzo nakon toga automat se počinje masovno proizvoditi i širiti na sve krajeve svijeta [1].

Danas se samoposlužni automati koriste na cijelom svijetu. Najveću ulogu imaju proizvodi iz prehrambene industrije, ali sve češće vidamo kako se pomoću njih mogu kupiti drugi proizvodi. Na samoposlužnim automatima može se kupiti gotovo sve, uključujući svježe pripravljenu pizzu i odjevne predmete, a u Sjedinjenim Američkim državama može uz skeniranje osobne iskaznice kupiti i streljivo za vatreno oružje. Iz navedenih primjera očito je da je primjena velika. U ovom radu usredotočit ćemo se na samoposlužni automat za prodaju manjih predmeta kao što su grickalice ili čokoladice, upoznat ćemo se s osnovnim dijelovima takvog uređaja te proučiti kako oni djeluju. Prikazat ćemo način djelovanja na jednostavnom primjeru gdje će se nakon ubacivanja kovanice i odabira željenog proizvoda moći kupiti određeni proizvod.

2. AUTOMATIKA I SAMOPOSLUŽNI AUTOMAT

Samoposlužni automat predstavlja automatizirani proces i time obuhvaća više struka iz područja tehničkih znanosti. Automatika je znanost (disciplina) koja se bavi proučavanjem, izradom, načinom upravljanja i održavanjem tehničkih uređaja koji obavljaju neku željenu aktivnost bez izravnog sudjelovanja čovjeka. Prema ovoj definiciji automat se smatra uređajem koji obavlja svoj rad bez prisustva čovjeka. Zadatak može biti jednostavan poput rada koji obavlja samoposlužni automat ili se može odnositi na cijela postrojenja ili njihove dijelove. Automatizacija tako predstavlja automatsko upravljanje strojevima, sustavima, uređajima i drugim procesima. korištenjem mehaničkih i elektroničkih uređaja, odnosno prelazak rada s čovjeka na stroj. Potreba za automatizacijom u današnje doba sve je veća zbog nedostatka radne snage i visokih troškova rada. Automatizacija omogućuje povećanu produktivnost, niže troškove, smanjenje vremena za izvođenje nekog procesa te naposljetku bolji proizvod ili uslugu. Iako automatika primarno spada pod tehničko polje elektrotehnike, da bismo automatizirali proces, potrebna su znanja iz više tehničkih disciplina. Potrebno znanje uključuje strojarstvo, elektrotehniku i računarstvo, a konačni cilj je stvaranje učinkovitog tehnološkog procesa. Može se očekivati kako će se te grane znanosti sve više razvijati u budućnosti te s njima i sve prethodno navedene discipline uz nove metode i različite sustave [2].

Samoposlužni automat je jedan od automatiziranih procesa te objedinjuje razne tehničke vještine zbog čega njegovim proučavanjem možemo mnogo toga naučiti. Razmjerno je jednostavnog dizajna i upravljanja, a uređaj je poznat gotovo svima i zato je prikladan za objašnjenje procesa automatizacije u cijelosti. Vidjet ćemo kako se električni elementi i napisani programski kod ponašaju zasebno i u cjelini koja čini uređaj, a također i proces izrade kućišta.

U ovom radu dotaknut ćemo se svih područja vezanih za automatizaciju nekog procesa. Primjerice, izrada kućišta je dio strojarstava, povezivanje električnih elemenata dio elektrotehnike, a programiranje dio računarstva. Na kraju rada, objasnit ćemo princip rada uređaja.

3. ELEKTROTEHNIKA

Elektrotehnika je znanost koja se bavi proučavanjem i primjenom električne energije u različitim disciplinama. Kako su za samoposlužni automat potrebni električni uređaji, u ovom poglavlju su navedeni i detaljno opisani.

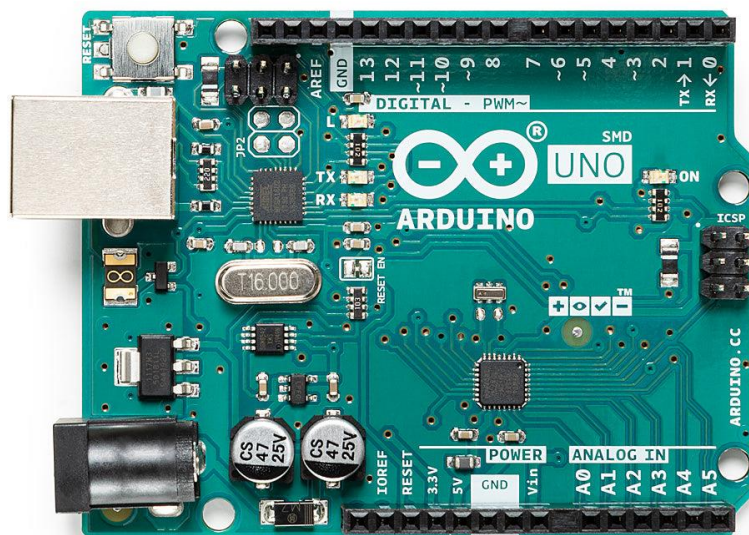
3.1. Arduino

Arduino je talijanska tvrtka koja je proizvela sklopovnlje i programsku podršku otvorenog tipa koji se mogu koristiti za različite vrste projekata koji uključuju električne uređaje. Tvrtka je osnovana 2005. godine kako bi se proizveo jednostavan uređaj pomoću kojeg svi jednostavno mogu upravljati električnim uređajima pomoću jedne od najjednostavnijih platformi. Navedena platforma prikladna je za sve koji se žele početi baviti mehatronikom ili izraditi projekt. Arduino je razvio pločice pomoću kojih se jednostavno upravlja elektoničkim komponentama kao što su motori, diode, senzori i drugi uređaji, nakon što smo programirali pločicu. Tvrtka je proizvela razne vrste pločica kao što su arduino Nano, Uno, Mega i druge. Zajedničkim imenom te pločice možemo nazvati Arduino pa se ime odnosi na tvrtku, ali i na pločice koje su osmislili. Programska podrška koja se koristi za pisanje koda naziva se Arduino integrirano okruženje za razvoj (eng. Integrated Development Environment (IDE)), koje je detaljnije opisano u ulomku 4.1 Arduino IDE. Najvažnija prednost ovog proizvoda je njegova niska cijena te besplatana programska podrška što doprinosi njegovoj popularnosti. Sklopovlje i programska podrška su otvorenog tipa što znači da su električne sheme pločica i izvorni kod javno dostupni svim korisnicima, a oni ga mogu izmijeniti kako bi postigli dodatnu funkcionalnost. Budući da postoji velik broj pločica različitih tvrtki, svaka sa posebnim karakteristikama, za svaki projekt možemo izabrati onu koja najbolje odgovara određenoj primjeni. Kako je izvorni kod javno dostupan, možemo ga modificirati da bi bolje odgovarao projektu, ali za to je potrebno dobro znanje programiranja [3].

Svaki Arduino sastoji se od mikrokontrolera (eng. microcontroller) koji se zasniva na tehnologiji integriranog kruga, a možemo ga zamisliti kao malo računalo, ali bez dodatne opreme nije upotrebljiv u praksi. Tvrtka Arduino upotrijebila je mikrokontroler i izradila tiskanu pločicu (eng. Printed Circuit Board, (PCB)) za njega. Pomoću tiskane pločice mehanički i električki se povezuju električne komponente. Tiskanom pločicom su povezali mikrokontroler sa ostalim električnim komponentama kako bi lakše upravljali mikrokontrolerima i takvu pločicu nazvali su

Arduino. Pomoću ulaza na tiskanoj pločici može se čitati ulazne podatke kao što su signal sa senzora ili pritisak tipke, a koristeći izlaze može se upravljati preko izlaza motorima, svijetlećim diodama i slično. Pločice mogu imati priključak za napajanje zavisno o modelu i možemo ih kablom spojiti na računalo kako bismo ih programirali. Navedene prednosti i niska cijena povećale su popularnost pločica koje se danas koriste u velikom broju.

Postoje različite inačice Arduino pločica, dizajnirane za različite projekte i sa različitim mogućnostima i performansama, mogu imati različit broj ulaza i izlaza, imati bežičnu (eng. Wireless Fidelity (Wi-Fi)) vezu i mnoge druge značajke. Arduino Nano pločica malih je dimenzija te je pogodna za projekte gdje je bitan prostor i možemo je smjestiti na eksperimentalnu pločicu, ali nema klasičnu vezu pomoću univerzalne serijske sabirnice (eng. Universal Serial Bus (USB)) konekciju niti napajanje. Manja verzija te pločice je Arduino Pro Mini koja ima najmanje dimenzije. Najveća pločica je Arduino MEGA koja ima najviše ulaza i izlaza, najviše memorije i najveću procesorsku snagu, pa je pogodna za zahtjevne projekte. Najpoznatiji i najčešće korišten je Arduino Uno koji je dimenzijama između prethodna dva modela, ima dovoljno veza za većinu projekata kao i mikrokontroler dovoljne snage i memorije. Postoji mnogo različitih pločica, svaka sa različitim karakteristikama, kao i pločice drugih proizvođača.



Slika 3.1. Arduino Uno R3 SMD

Još jedna zanimljiva značajka Arduina je Arduino Shield, prikazan na slici 3.2. To je dodatni dio sklopovlja koji priključujemo na vrh Arduina na njegove priključke (eng. headers) kako bi dobili

nove mogućnosti kao npr. Shield u koji je ugrađen Wi-Fi za povezivanje s internetom ili Shield za jednostavnije upravljanje motorima ako ih koristimo mnogo, ali u ovom projektu nije korišten.



Slika 3.2. Arduino Shield povezan na Arduino

3.1.1. Arduino Uno R3 SMD

U ovom projektu korišten je Arduino Uno R3 SMD pločica. Pločica Arduino Uno ima najbolji omjer procesorske snage, memorije, broja ulaza i izlaza i sadrži potrebne značajke za jednostavan rad prilikom izrade samoposlužnog automata. Dimenzije su mu 68.6 x 53.4 mm, a težina mu je 25 grama. Arduino je središnji dio projekta na koji ćemo priključiti sve električne elemente, a on će pomoću napisanog koda njima upravljati. R3 u imenu označava najnoviju treću verziju Arduino Uno pločice. SMD znači da je mikrokontroler integriran u tiskanu pločicu te ga ne možemo izvaditi. Naziv Arduino Uno R3 znači da se mikokontroler može lako zamijeniti u slučaju oštećenja. Opisana značajka je jedina razlika u odnosu na prethodne modele. Korišteni model je prikazan na slici 3.1. Radni napon mu iznosi 5 V. Možemo ga napajati preko USB kabela, posebnog konektora ili jednog od pinova. Napajanjem preko USB kabela Arduino izravno dobiva potrebnih 5 V i ograničen je na struju do 0.5 A. Arduino koristi oko 50 mA, a ostatak struje koristi se za povezane komponente. Kada ga napajamo preko konektora ili nekog od pinova, struja prvo prolazi kroz ugrađeni pretvarač napona koji pretva napon na potrebnih 5 V, a struja koji koristi je najviše 1 A. Maksimalni napon kojim će regulator napona moći upravljati je između 6 i 20 V, ali preporučeno je koristiti izvor napajanja između 7 i 12 V. Preko

USB veze moguće je jednostavno povezivanje s računalom i osim za napanje Arduina koristimo ga i za njegovo programiranje, odnosno prijenos koda sa računala na njega. Arduino ima tipku za ponovno pokretanje koja se koristi da ponovno pokrenemo Arduino, učitamo kod na njega ili pokrenemo postojeći kod [4].

Mikrokontroler kojeg ovaj Arduino koristi je ATmega328. Ovaj mikrokontroler sadrži jedan integrirani krug ili čip (eng. chip) od tvrtke Atmel. Za rad mu je potrebno između 1.8V i 5.5V. Neke od karakteristika su prikazane u tablici 3.1. Arduino ima 32 KB flash memorije od kojih 0.5 KB koristi pokretački program (eng. bootloader). Pokretački program je dio programske podrške pomoću kojeg se napisani kod učitava na mikrokontroler. Mikrokontroler je tako najvažniji dio pločice i on je taj koji obavlja zadane programske zadatke [5].

Tablica 3.1. Osnovne karakteristike čipa

Tip procesora	8-bitni tip središnje jedinice (procesora) sa smanjenim skupom naredaba (eng. Reduced Instruction Set Computer (RISC))
Maksimalna brzina procesora	20 MHz
Flash memorija	32 KB
Statička memoria s nasumičnim pristupom (eng. Static Random-Access Memory (SRAM))	2 KB
Električno izbrisiva programibilna ispisna memorija (eng. Electrically Erasable Programmable Read-Only Memory (EEPROM))	1 KB
Maksimalan broj ulaza i izlaza	23
Vanjski prekidi	3
Broj pinova	32

Pločica ima 14 digitalnih pinova koji se mogu ponašati kao ulazi ili izlazi, od kojih 6 može biti korišteno kao izlazi za modulaciju širine impulsa (eng. Pulse Width Modulation (PWM)), a također ima 6 analognih ulaza. Na Arduinu se nalazi 32 pina koji se mogu koristiti. U tablicama 3.2., 3.3., 3.4., 3.5. i slici 3.3 objašnjeno je značenje svih pinova. Pinove ćemo radi lakšeg snalaženja podijeliti u četiri skupine na pinove snage, analogne, digitalne i ostale pinove [6].

Tablica 3.2. Pinovi snage

Oznaka pina:	Opis:
NC	Nije električki povezan s pločicom.
IOREF	Daje referentni napon s kojim pločica radi.
RESET	Ponovno pokretanje pločice.
3.3V	Služi za napajanje komponenata iz pločice od 3.3 V.
5V	Služi za napajanje komponenata iz pločice od 5 V.
GND	Uzemljenje (eng. ground), predstavlja referentni napon od 0 V na Arduinu.
Vin	Ovaj pin možemo koristiti za napajanje Arduina tako što ćemo na ovaj pin spojiti pozitivan pol napajanja napona između 7 V i 12 V, a negativan pol na GND pin.

Tablica 3.3. Analogni pinovi

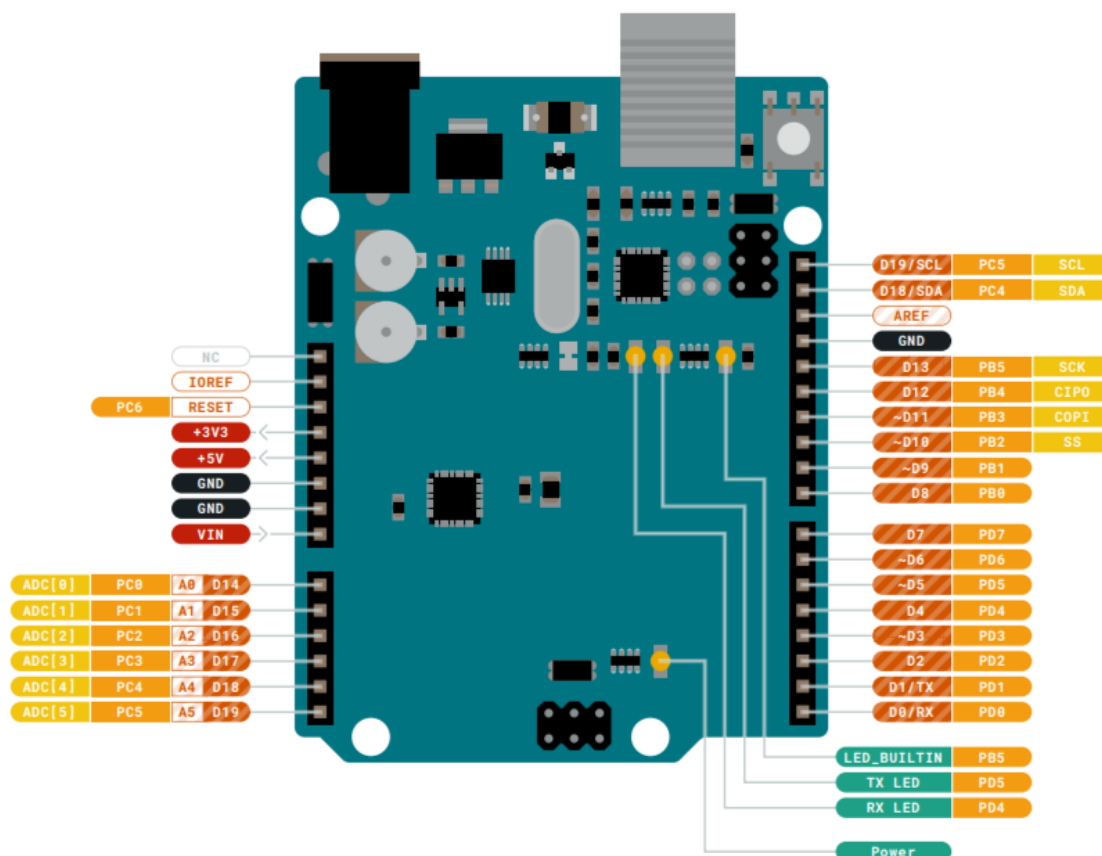
Oznaka pina:	Opis:
A0	Predstavljaju 6 analognih ulaza u Arduino koji mogu čitati vrijednosti napona između 0 V i 5 V koje pretvaraju pomoću analogno-digitalnog (engl. Analog-to-Digital (A/D)) pretvarača u digitalni zapis između 0 i 1023 bita.
A1	
A2	
A3	
A4	
A5	

Tablica 3.4. Digitalni pinovi

Oznaka pina:	Opis:	
D13	Ovi pinovi predstavljaju digitalne ulaze ili izlaze pomoću kojih upravljamo ili uzimamo informacije s drugih komponenata odnosno njihov napon. Imaju dva stanja, nisko (eng. low), koje je predstavljeno naponom od 0 V, ili visoko (eng. high) stanje, koje je predstavljeno naponom od 5 V na pinu. Na njih možemo gledati kao na binarne pinove jer poprimaju isključivo vrijednosti 0 ili 1 kada na njih djeluje 0 ili 5 V. Prije korištenja moramo postaviti pin u stanje u kojem želimo da radi (kao ulaz ili kao izlaz). Ako ga postavimo kao ulaz on će čitati stanje napona na njemu, a ako ga postavimo kao izlaz on će slati napon. Na nekim pinovima vidimo znak ~ (tilda) što znači da se ti pinovi mogu koristiti za PWM. Maksimalna struja koja se može koristiti preko ovih pinova iznosi 40 mA, a poželjno bi bilo da iznos oko 20 mA	
D12		
~D11		
~D10		
~D9		
D8		
D7		
~D6		
~D5		
D4		
~D3		
DD2		
D1/TX		Ove pinove možemo koristiti kao digitalne ulaze/izlaze ili kao pinove za UART (eng. Universal Asynchronous Receiver / Transmitter (UART)) protokol. UART je serijski komunikacijski protokol pomoću kojeg pločica komunicira sa računalom ili drugim pločicama.
D0/RX		

Tablica 3.5. Ostali pinovi

Oznaka pina:	Opis:
SCL	Ove pinove koristimo za I2C (eng. Inter-Integrated Circuit (I2C)) komunikacijski protokol između pločice i određene komponente. Na isti način možemo koristiti pin A4 umjesto SDA i pin A5 umjesto SCL, ali ne možemo ih koristiti u isto vrijeme.
SDA	
AREF	Predstavlja referentni analogni napon koji možemo koristiti tijekom A/D pretvorbe.



Slika 3.3. Prikaz pinova na pločici Arduino Uno R3 SMD

Pinovi D13/SCK, D12/MISO, D11/MOSI i D10/SS mogu se koristiti za SPI (eng. Serial Peripheral Interface) protokol. Koristimo ga za brzu komunikaciju između pločice i određene komponente, ali dužina žice ne smije biti veća od nekoliko centimetara zato što podaci koji se šalju na većim udaljenostima postanu izobličeni.

Na pločici su ugrađene četiri svjetleće diode. Jedna od njih prikazuje je pločica uključena ili isključena. Druga je priključena na pin TX i treperi kada mikrokontroler šalje podatke. Dioda

priključena na klin RX treperi kada prima podatke. Na pin D13 ugrađena je svjetleća dioda koja svijetli kada je pin u visokom stanju, a kada je u niskom ne svijetli.

Neki od nedostataka Arduina su niska razina preciznosti, mali broj komunikacijskih protokola, mala procesorsku snagu i memorija te ograničene sigurnosne značajke. Zbog navedenih mana nije korišten kao trajno rješenje u velikim projektima.

3.2. Servomotor

Servomotor je električni motor koji pretvara električnu energiju u mehaničku energiju na vlastitoj osovini koja obavlja koristan rad. Koristi istosmjernu struju pomoću koje stvara okretni moment ili silu. Servo motor je poseban po tome što se može zaustaviti u određenom položaju dovođenjem upravljačkog signala. Položaj može biti zakretni položaj kod rotacijskog servomotora ili položaj na putanji kod linearnog servomotora. Servo motorima upravlja se pomoću regulacijskog kruga koji upravljački signal pretvara u pomake. Primjene ovakvih motora su široke i postoje različiti modeli zavisno o potrebi, a koriste se u industrijskim primjenama i jednostavnim uređajima. Rad s njima je jednostavan i brzo se može svladati način na koji se njima upravlja [7].

Usredotočit ćemo se na servomotore koji se mogu koristiti zajedno s Arduino pločicama. Mali su, masovno se proizvode i cjenovno su pristupačni. Neki od najpoznatijih modela su MG90S, SG90, FS90R i drugi. Prikladni su za rad s arduino pločicama za izradu različitih projekata. Male su težine i dimenzija. Pozicijski servomotor može se zaustaviti precizno u bilo kojem položaju između 0 i 180 stupnjeva dovođenjem upravljačkog signala koji kontrolira za koliko će se osovina motora zakrenuti. Postoje i servomotore koji imaju opseg između 0 i 270 stupnjeva ili 0 i 150 stupnjeva. Prikladni su za pozicioniranje senzora, kamera i drugih uređaja. Ovakav motor ne može napraviti puni okret i naziva se servomotor. Servomotori sastoje se od istosmjernog motora, potencijometra, kontrolne ploče i reduktora koji su zajedno smješteni u kućištu. Imaju tri izlazna pina od kojih se dva koriste za napajanje, a treći kao signal za upravljanje položajem. Ovisno o proizvođaču pin crne ili smeđe boje je GND pin, crveni pin je snaga, a žuta ili narančasta boja označava pin za signal. Servomotor pokreće se pomoću istosmjernog (eng. Direct Current, (DC)) motora, potencijometar daje signal o poziciji osovine u svakom trenutku, kontrolna ploča povezana je s potencijometrom i ona upravlja za koliko će se servomotor okretati dok osovina ne dođe u željeni položaj pomoću PWM-a, što je detaljno opisano u podulomku 3.2.1. Tower Pro MG995 360° koji opisuje korišteni servomotor. Reduktor se sastoji od

zupčanika koji povećavaju broj okretaja na osovini u odnosu na broj okretaja DC motora kako bi dobili veći moment. Za primjene gdje nam je potrebna puna rotacija servomotora koristimo kontinuirani servomotor koji se može okrenuti za svih 360 stupnjeva, ali ne može se precizno pozicionirati. Za izradu samoposlužnog automata koristiti ćemo kontinuirane servomotore.

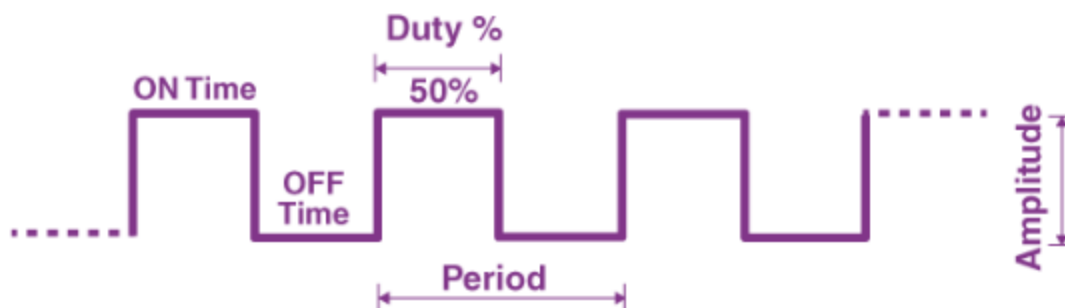


Slika 3.4. Servomotor Tower Pro MG995 360°

3.2.1. Tower Pro MG995 360°

U ovom projektu koristit ćemo Tower Pro MG995 360° servomotor. Taj model je kontinuirani servo motor prikladan za korištenje zajedno sa Arduino Uno pločicom. Za razliku od pozicijskog servomotora, ovaj model nema ugrađeni potencijometar zato što nije potrebno precizno pozicioniranje. Ovaj motor sastoji se od DC motora, kontrolne ploče i reduktora. Ima tri pina od kojih dva služe za napajanje motora i treći za signal. Za ovaj projekt nije važna težina ili veličina servomotora kao što bi to bio slučaj da koristimo, na primjer, bespilotne letjelice (eng. drone), pa su nam najvažnije značajke za odabir njegova brzina i moment. Dimenzije motora su 20 x 40 x 43 mm i teži 55 g. Brzina se prikazuje pomoću vremenu koje je potrebno da se osovina okrene za 60 stupnjeva. Moment nam govori koliku silu servomotor može primijeniti na polugu i izražava se u kg/cm. Ta brojka znači da će se na jednom centimetru udaljenosti od osovine motor moći suprotstaviti naznačenoj masi, dva centimetra udaljenosti to će se smanjiti za pola, onda za četvrtinu i dalje prema istom načelu. Brzina i moment ovise o spojenom naponu. Ako motor spojimo na minimalni napon od 4.8 V brzina će mu biti 0.17 s/60°, a moment 13 kg/cm. Ako motor spojimo na maksimalni napon od 6V brzina će mu biti 0.13 s/60°, a moment 15 kg/cm. Za napon između tih vrijednosti mijenjati će se i karakteristike motora između napisanih vrijednosti.

Modulacija širine impulsa, skraćeno PWM (eng. Pulse Width Modulation) je metoda kojom možemo upravljati pravokutnim valovima tako što ćemo im smanjivati ili povećavati širinu. Ovu metodu koristimo za slanje željene snage nekom električnom uređaju iz konstantnog izvora. Na ovaj način možemo mijenjati svjetlinu svjetleće diode, upravljati brzinom motora i slično. Ovom metodom možemo isključivati i uključivati istosmjerni napon tako da on više ne bude konstantan, već dobivamo pravokutni signal koji se određeno vrijeme nalazi u niskom stanju, a ostatak vremena u visokom stanju, čime se postiže pravokutni oblik napona. Radni ciklus (eng. duty cycle) koji prikazujemo u postocima nam govori koliki udio vremena signal provodi u niskom, a koliki u visokom stanju tijekom jednog perioda. Možemo ga prikazati i vremenski, a u tom slučaju vrijeme pokazuje koliko dugo je visoko stanje bilo aktivno tijekom jednog perioda. Primjerice, napon iznosi 5 V uz radni ciklus od 20 posto, efektivna vrijednost na izlazu iznosi 1 V zato što će tijekom jednog ciklusa 20 posto vremena biti prisutan napon od 5 V (visoko stanje), a ostalih 80 posto biti će 0 V (nisko stanje). Ovom metodom promjenom širine pravokutnog impulsa možemo slati drugačije vrijednosti napona u električni krug, što predstavlja signal za upravljanje. Slika 3.5. prikazuje primjer pravokutnog signala uz oznake za ON Time (visoko stanje), OFF Time (nisko stanje) i Duty % (radni ciklus) od 50% (signal se jednako dugo nalazi u niskom i visokom stanju) [8].



Slika 3.5. Prikaz modulacije širine impulsa

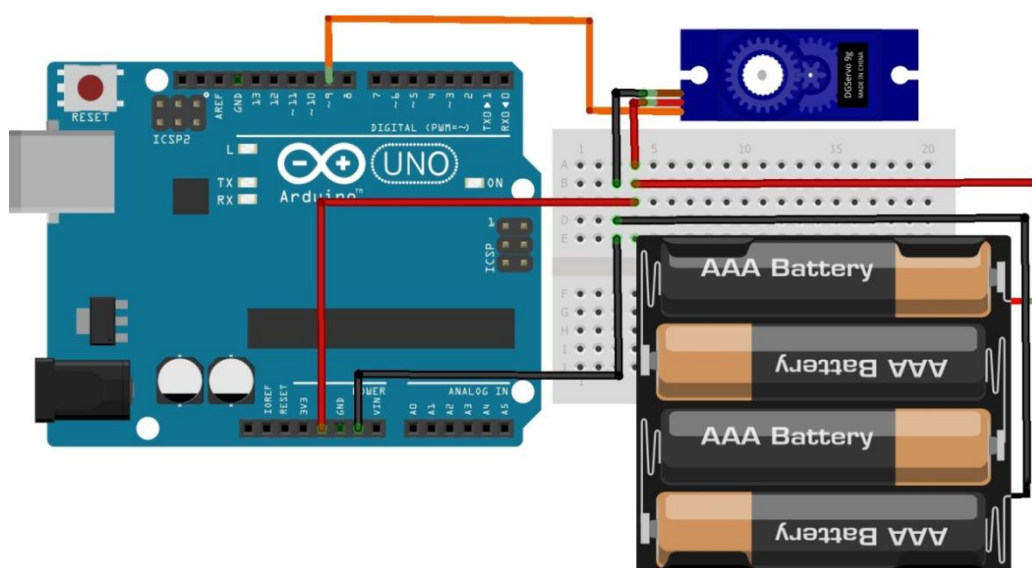
Kod pozicijskih servomotora pomoću PWM-a kontroliramo signal koji dobiva motor te se ovisno o njemu osovina pozicionira na željenu poziciju. Kontroliranjem širine pravokutnog oblika, odnosno vremena trajanja visokog stanja kontroliramo u kojem položaju između 0 i 180 stupnjeva će se pozicijski servomotor zaustaviti, ovisno o motoru. Kontinuiranim servomotorom upravljamo na isti način i uz istu širinu impulsa, ali uz važnu razliku. Kod kontinuiranih servomotora ne upravljamo položajem osovine, već smjerom okretanja, kao što je prikazano u tablici 3.6. Nije bitan broj impulsa koji se šalje motoru, već njihova širina, odnosno vrijeme trajanja impulsa kojim postizemo željeni zaokret.

Tablica 3.6. Usporedba pozicijskih i kontinuiranih servo motora

	Pozicijski, 180°	Kontinuirani, 360°
Širina impulsa u trajanju od 1.5 ms, broj 90 u programskom kodu.	Osovina se nalazi na polovici vlastitog opsega, na 90 stupnjeva.	Stanje mirovanja u bilo kojoj poziciji između 0 i 360 stupnjeva.
Širina impulsa u trajanju od 1 ms, broj 0 u programskom kodu.	Osovina se okreće do krajnje točke od 0 stupnjeva.	Osovina se okreće suprotno od smjera kazaljke na satu punom brzinom.
Širina impulsa u trajanju od 2 ms, broj 180 u programskom kodu.	Osovina se okreće do krajnje točke od 180 stupnjeva.	Osovina se okreće u smjeru kazaljke na satu punom brzinom.

Ako u nekom projektu koristimo mnogo servomotora ili želimo kontrolirati svjetleće diode, može se koristiti PCA9685 modul koji nam omogućuje da kontroliramo do 16 servomotora ili LED zaslona istovremeno pomoću I2C komunikacijskog protokola dok je bez njega sa Arduino Uno moguće kontrolirati najviše 6 zato što toliko imamo PWM pinova. U ovom projektu nećemo ga koristiti zato što koristimo samo dva servomotora.

Servomotor moguće je napajati direktno iz Arduino Uno pločice preko 5 V pina, ali to nije preporučeno pogotovo za veće motore. Motor prilikom rada povlači dovoljno veliku struju da bi Arduino mogao ostati bez nje, a tada bi se Arduino stalno ponovno pokretao i mogao bi se oštetiti. Zato je najbolje napajati motore iz posebnog napajanja koje će napajati samo njih. U primjeru ćemo spojiti servomotor prema shemi na slici 3.6.



Slika 3.6. Shema spoja servomotora sa Arduinoom

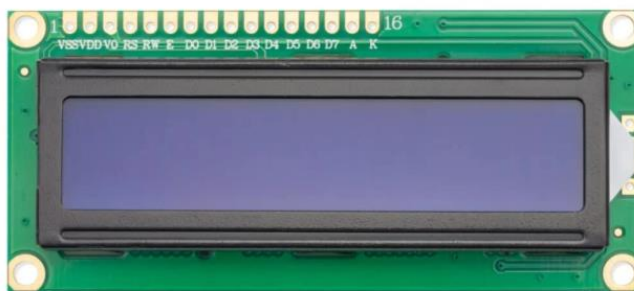
Servomotor je spojen na posebno napajanje od 6 V i na GND pin Arduina kako bi imali isti referentni napon od 0 V. Nakon što smo našem servomotoru omogućili napajanje preostali pin spajamo sa Arduinoom na jedan od digitalnih pinova, ali moramo paziti da to bude pin koji pored sebe ima znak tilde (~) koji označava da se radi o pinu koji podržava PWM.

3.3. LCD

LCD (eng. liquid crystal display) je električni zaslon korišten u električnim uređajima na kojem možemo prikazivati riječi, slova, znakove ili slike. U uporabi su već dug niz godina na zaslonima digitalnih satova i kalkulatora do mobitela, monitora i televizijskih prijemnika. Kada su izumljeni zamijenili su dotadašnje zaslone, koji su koristili katodne cijevi, u gotovo svim primjenama. Zaslone s katodnim cijevima bili su znatno veći i trošili više energije. LCD ekrani se sastoje od stotina piksela koji mogu propuštati svjetlo ili ne propuštati svjetlo. Najbitniji dio piksela su molekule tekućih kristala po kojima je ekran nazvan. Tekući kristali nalaze se između prozirnih elektroda i polariziranih filtara od kojih je jedan okomito orijentiran, a drugi paralelno. Prije nego se dovede napon tekući kristali imaju uvrnuti oblik i ekran djeluje sivo, a dovođenjem napona i stvaranjem električnog polja tekući kristali poprimiti će drugi oblik zbog čega će blokirati svjetlost, a piksel će izgledati kao da je crne boje. Dovođenjem ili izostankom napona neki pikseli će propuštiti svjetlo, a neki ne što će u kombinaciji sa stotinama piksela omogućiti da vidimo sliku na ekranu. Tekući kristali ne emitiraju svjetlost, a zato je potrebno imati pozadinsko svjetlo za koje se najčešće koriste svjetleće diode [9].

3.3.1. LCD 16x2

Za ovaj projekt korišten je zaslon LCD 16x2 zaslon. To je svestran zaslon primjenjiv za ispisivanje teksta, znakova, brojeva i drugih simbola. Troši jako malo energije i nije skup. Radi se o zaslonu koji ima 16 redova i 2 stupca i zato je u naslovu navedeno 16x2. Zaslon može u jednom trenutku ispisati do 32 različita znaka. Svaki od znakova ispisuje se na zaslonu pomoću 40 različitih piksela što znači da ih ovaj zaslon ima ukupno 1280 i svaki od njih sastoji se od zasebnog tekućeg kristala i radi prema prije objašnjenom principu. Ima plavo pozadinsko svjetlo i bijele znakove. Za rad mu je potrebno 5 V. Dimenzije su mu 80 x 36 x 12 mm i dimenzije samog zaslona su 66 x 16 mm.



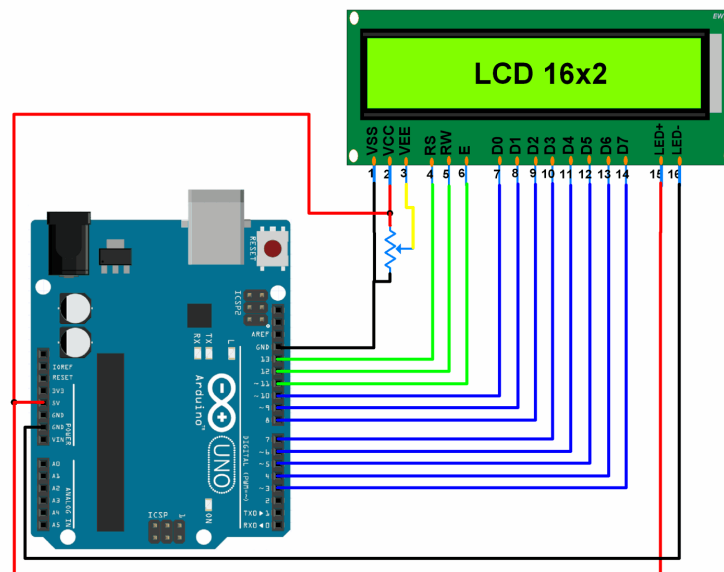
Slika 3.7. LCD 16x2

Ovakav zaslon primjenjiv je za jednostavan rad i povezivanje s Arduino pločicama kao i za njihovo programiranje. Na slici 3.7. možemo vidjeti da se sastoji od 16 različitih pinova koji su potrebni za njegov rad i opisani u tablici 3.7.

Tablica 3.7. Oznaka i opis pinova

Oznaka pina:	Opis:
VSS	Označuje GND pin.
VDD	Pin na koji spajamo napajanje odnosno 5 V.
V0	Kontrolira svjetlinu zaslona spajanjem preko potencijometra.
RS	Pin koji omogućava odabir registra (eng. register select pin) koji kontrolira gdje se u LCD memoriji zapisuju podaci. Možemo izabrati podatkovni registar koji zadržava ono što se prikazuje na zaslonu ili instrukcijski registar pomoću kojeg LCD dobiva instrukcije što da prikazuje sljedeće.
RW	Označava pin za čitanje/pisanje (eng. read/write) pomoću kojeg odabiremo način rada, odnosno želimo li zapisati podatak na zaslon ili očitati podatak sa zaslona.
E	Označava pin koji omogućava (eng. enable) pisanje podataka u registar.
D0	Ovi pinovi su paralelni pinovi za unos podataka što znači da se pomoću njih jedan po jedan bit podatka prenosi iz Arduina na LCD preko svakog pina (komunikacijske linije). Postoje dva načina unosa podataka, a to su 8-bitni način i 4-bitni način. 8-bitni način koristi svih 8 pinova za unos podataka, a 4-bitni način koristi 4 pina što smanjuje broj žica koje moramo povezati na Arduino, ali tada je potrebno više transakcija između Arduina i LCD-a za prijenos istog broja bitova.
D1	
D2	
D3	
D4	
D5	
D6	
D7	
A	Označava anodu i povezuje se na pozitivan pol od 5 V, zajedno sa sljedećim pinom omogućuju pozadinsko osvjetljenje pomoću svjetlećih dioda.
K	Označava katodu i povezuje se na negativan pol od 0 V, odnosno GND.

Na slici 3.8 prikazan je spoj Arduina sa LCD-om u 8-bitnoj konfiguraciji.



Slika 3.8. Arduino Uno povezan sa LCD-om

Možemo primijetiti kako za spajanje zaslona sa Arduinoom moramo koristiti čak 16 žica i iskoristiti velik broj digitalnih pinova. To može biti dobro rješenje ako u projektu koristimo samo zaslon. Ako namjeravamo spojiti više električnih elemenata ili jednostavno smanjiti broj žica možemo koristiti I2C komunikacijski protokol.

IIC ili I2C (eng. Inter-Integrated Circuit) je komunikacijski protokol kojeg koristimo ako želimo ostvariti brzu i jednostavnu komunikaciju između električnih uređaja. Ovakva komunikacija je smišljena 1982. godine od tvrtke Philips Semiconductors za povezivanje televizijskih prijemnika i mikrokontrolera. Spajanje se jednostavno postiže pomoću žica i koristi serijsku komunikaciju. Pomoću serijske komunikacije prenosi se jedan po jedan bit preko jedne komunikacijske linije. U odnosu na paralelnu komunikaciju serijska je jednostavnija, pouzdanija i pogodnija za veće udaljenosti. Može se koristiti za povezivanje integriranih krugova i računala, možemo povezati više Arduina međusobno kao i razne električne komponente s mikrokontrolerima, a zbog toga je vrlo korisna i često korištena. Uobičajena brzina komunikacije je 100 kbit/s u standardnom načinu rada ili 400 kbit/s u brzom načinu, a može ići sve do 5 Mbit/s [10, 11].

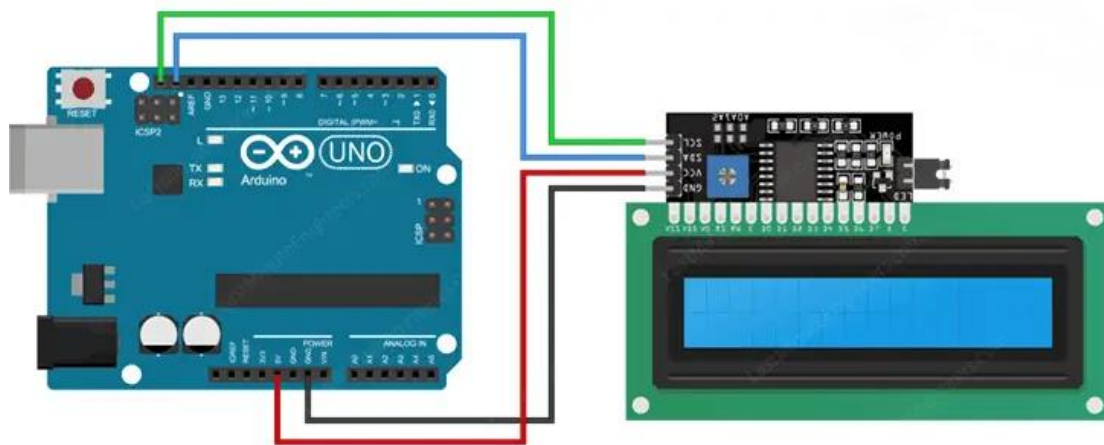
Za spajanje Arduino pločice i LCD-a pomoću I2C protokola koristit ćemo I2C modul prikazan na slici 3.9. To je dio sklopovlja pomoću kojeg ostvarujemo komunikaciju. Modul ima 16 pinova koje povezujemo na 16 pinova LCD-a, a može se i kupiti LCD sa već zalemljenim modulom. Modul se sastoji od četiri izlazna pina koji su upisani u tablici 3.8. i koji se koriste za spajanje

I2C komunikacije sa Arduinoom i tako je smanjen broj potrebnih veza s Arduinoom na samo četiri. Na modulu se još nalazi kontrola za prilagodbu svjetline i mogućnost promjene adrese ako koristimo više I2C konekcija. Od četiri izlazna pina dva se koriste za napajanje, a druga dva su SDA i SCL. SCL predstavlja signal sata (eng. clock signal), a to u digitalnim krugovima predstavlja signal koji oscilira između niskog i visokog stanja konstantnom frekvencijom kako bi sinkronizirao radnje digitalnih elemenata. SDA konekcija koristi se za serijski prijenos podataka jedan po jedan bit. Svakim prelaženjem signal sata iz visokog u nisko stanje ili obrnuto šalje se jedan bit između odašiljača i prijemnika sve dok nije poslano 7 bitova koji čine adresu, u ovom slučaju naredbu ili dio informacije. Preko istog pina SDA šaljemo i pišemo informacije između komponenata i zato postoje četiri različita stanja. Odašiljač i prijemnik mogu biti i LCD i Arduino ovisno tko šalje a tko prima informacije. Arduino i LCD su u stanju odašiljača kada šalju podatke i oba mogu biti u stanju prijemnika kada primaju podatke. Njihove uloge se mijenjaju tijekom komunikacije i nije moguće da se oba nađu u istom stanju. Uz slanje 0 ili 1 ova komunikacija omogućuje i slanje posebnog START ili STOP signala koji čine granicu poslanih informacija. Kada odašiljač želi komunicirati s prijemnikom, pošalje START signal nakon čega pošalje 7 bitnu adresu, a sljedeći bit označava želi li pisati (nula) ili čitati (jedan) sa prijemnika. Prijemnik će odgovoriti sa ACK bitom koji služi kao bit za potvrdu da je primio podatke i postupak se ponavlja sve dok se ne prenese cijela informacija [10].

Tablica 3.8. Oznake pinova na I2C modulu

Naziv pina:	Opis:
VCC	Napajanje, spajamo sa 5 V pinom Arduina.
GND	Referentni napon od 0 V spajamo na GND pin Arduina.
SDA	Pin za prijenos podataka koji spajamo na SDA pin na Arduino ili na analogni pin A4.
SCL	Ovaj pin označuje signal sata i spajamo ga na SCL pin Arduina ili na analogni pin A5.

Možemo primijeniti da kako ovakav način spajanja ne koristi niti jedan od digitalnih pinova što je korisno zato što će nam oni biti potrebni za spoj ostalih komponenata.



Slika 3.9. Arduino Uno povezan sa LCD-om preko IC2 modula

U ovom projektu korišten je I2C modul za povezivanje Arduino pločice i LCD-a kako bi nam preostalo dovoljno konekcija za ostale komponente.

3.4. Tipkovnica

Tipkovnica je skup tipki koje se koriste za slanje signala nekom električnom uređaju. Pritiskom na tipku zatvara se ili otvara električni krug koji šalje signal uređaju za obavljanje neke radnje. Postoje različite vrste tipkovnica za primjene u kojima je važno da su tipke organizirane u jednu smislenu cjelinu. Radi praktičnosti tipke najčešće imaju jedinstvenu funkciju, ali mogu imati i više funkcija. Komunikacija tipkovnicom je jedan od najjednostavnijih oblika komunikacije sa električnim uređajima. Tipkovnice se koriste kao ulazne jedinice i njima isključivo šaljemo signal električnom uređaju, a ne primamo ih. One su fizički predmeti, ali mogu biti i digitalne poput tipkovnice pametnih mobitela. Tipkovnicu čine tipke sa oznakama brojeva, slova, simbola, a mogu i predstavljati neku funkciju ili naredbu uređaju. Zastupljene su u gotovo svim sferama i jednostavne su za izradu. Koristimo ih za unos podataka u računalo, numeričke tipkovnice koristimo za unos podataka u kalkulator, bankomat, daljinski upravljač i druge uređaje.

3.4.1. 3x4 Matrix Membrane Keypad

U ovom projektu korišten je 3x4 Matrix Membrane Keypad. To je tipkovnica napravljena za rad s Arduino pločicama. Ima potrebne konekcijske pinove i jednostavno je upravljati njome. Koristi membranske tipke raspoređene u x-y matrici koja se ne sastoji od odvojenih pokretnih dijelova kao kod većine ostalih tipkovnica već koristi jastučice osjetljive na pritisak. Ti jastučići koriste samo obrise tipki i simbole ispisane na ravnoj fleksibilnoj površini, pa se prilikom pritiska tipke

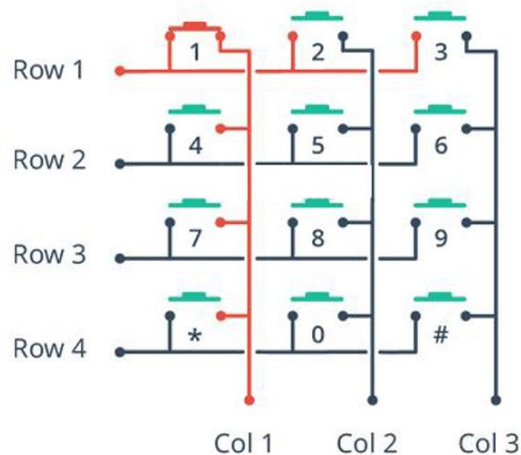
osjeća vrlo malo taktilne povratne informacije što ih može činiti težima za korištenje. Koriste električne kontakte na površini koji prilikom pritiskanja jastučića ostvaruju kontakt sa strujnim krugom ispod njih. Ovakve tipkovnice su jeftine i otporne na vodu i prljavštinu više nego ostale. Korisne su za izgradnju različitih projekata u kojima je potreban korisnički unos podataka, najčešće numeričkih, u primjenama kao što je odabir izbornika, unos lozinke i druge situacije [12].

Ova tipkovnica ima 12 tipki raspoređenih u 3 reda i 4 stupca, a često je korištena inačica 4x4 koja ima 16 tipki, kao što je vidljivo na slici 3.10. Rad s njima je gotovo identičan potrebne su male promjene tijekom pisanja koda. Tipke predstavljaju brojeve od 0 do 9 kao i znak za ljestve i zvjezdicu. Dimenzije su joj 68.9 x 76 x 0.8 mm i teži samo 7 grama.



Slika 3.10. 3x4 i 4x4 membranska tipkovnica

Ispod svake tipke nalazi se membranski prekidač koji, kada je tipka pritisnuta, električki povezuje red i stupac u kojem se nalazi, kao na slici 3.11. Prije nego što pritisnemo tipku svi stupci se ponašaju kao ulazi i postavljeni su u visoko stanje, odnosno logičku jedinicu, a svi redovi se ponašaju kao izlazi i postavljeni su u nisko stanje, logičku nulu. Pritiskom na tipku zatvara se strujni krug između reda i stupca i poteče struja. Pošto se stupci ponašaju kao ulazi, u stupcu gdje je pritisnuta tipka promijenit će se stanje iz visokog u nisko. Nakon toga Arduino zna u kojem stupcu je pritisnuta tipka. Kako bi znao u kojem redu je pritisnuta, on postavlja stanje svih stupaca u nisko i počinje svaki red prebacivati iz niskog u visoko stanje sve dok ne dođe do reda gdje je pritisnuta tipka. Tada će stupac doći u visoko stanje i Arduino će znati točan red i stupac u kojem je pritisnuta tipka. Arduino prepoznaje samo jedan pritisak tipke u određenom trenutku [13].



Slika 3.11. Način povezivanja redova i stupaca

Tipkovnica ima 7 pinova za povezivanje s Arduino pločicom. Svaki od njih predstavlja red ili stupac. Prva 4 od njih predstavljaju broj redova (eng. row) i ostala 3 broj stupaca (eng. column). Povezivanje je vrlo jednostavno na način da naših 7 pinova spojimo na digitalne izlazno/ulazne pinove Arduina.

3.5. Senzori

Senzori su električni uređaji koji mjere fizikalnu veličinu iz okoline i pretvaraju ju u električni signal. Oni detektiraju promjenu mjerene fizikalne veličine i šalju signal o toj promjeni. Dio su mjernog sustava koji je u izravnom dodiru s mjernom veličinom koju želimo izmjeriti. Sustav se može sastojati od samo jednog ili od nekoliko tisuća senzora. Mjerni sustav na ovaj način mjeri točnu ulaznu vrijednost fizikalne veličine kao što je temperatura, tlak, protok, sila i druge. Pomoću senzora koji detektiraju promjenu fizikalne veličine šalje se signal, najčešće računalu, koje će izraziti taj signal u formatu razumljivom čovjeku u obliku izlazne vrijednosti. Senzore možemo podijeliti u različite grupe, prema tipu mjerne veličine, načinu pretvorbe, materijala, području primjene i na druge načine. Svaki od njih ima različite karakteristike od kojih su neke mjerno područje, raspon, pogreška, rezolucija ili linearnost što znatno utječe na njihovu cijenu ovisno o potrebama primjene. Senzori su široko zastupljeni u današnjem svijetu, u različitim primjenama poput senzora otiska prsta na pametnom telefonu te senzora u roverima za istraživanje svemira. Mogu biti skupi kod iznimno preciznih i kompleksnih mjerenja ili vrlo jeftini. Koriste se u svim granama industrije poput kućanstva, medicine, vojske, avio industrije ili u slobodnoj primjeni. Senzori mogu poslužiti za detekciju zvuka, promjenu temperature, mjerenje pozicije, razine fluida ili pomaka [14].

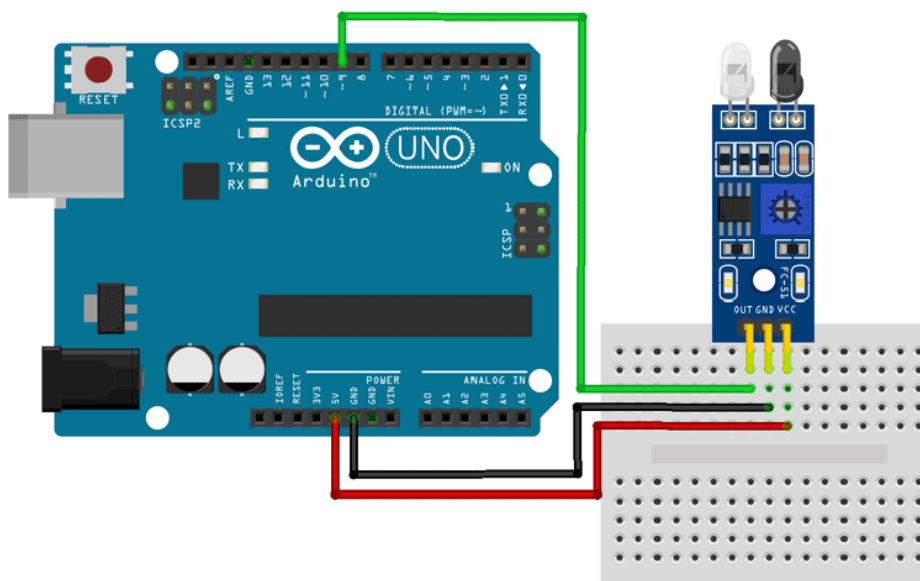
3.5.1. FC-51 infracrveni senzor blizine

Model senzora koji je korišten u ovom projektu je FC-51 IR senzor, prikazan na slici 3.12. To je senzor diskretne pozicije (eng. proximity sensor), koji se naziva i senzorom blizine, a koristi infracrveno svjetlo. Radni napon senzora je između 3 V i 6 V. Dimenzije su mu 45 x 14 x 7 mm. Jednostavno ga je koristiti za rad s Arduino pločicama.

Senzori diskretne pozicije su senzori koji mogu detektirati nalazi li se ili ne u njihovoj blizini neki predmet. Ova vrsta senzora je beskontaktni senzor pa ne mora ostvariti fizički kontakt s predmetom kako bi ga detektirao. Zbog toga imaju dug vijek trajanja, ali vijek trajanja je dulji također zbog toga što nemaju pokretne mehaničke dijelove. Senzori diskretne pozicije su jedni od najčešće korištenih senzora koje srećemo u svakodnevnom životu. Druge vrste ovih senzora koriste drugačije načine detekcije objekta. Primjerice, magnetski senzori koriste magnet, kapacitivni kondenzator i induktivni zavojnicu. Senzore koji koriste svjetlo nazivamo optički senzori i u njih spada senzor kojim se bavimo [15].

Optički senzori sastoje se od odašiljača i prijemnika koji se nalaze na modulu zajedno s upravljačkim sklopom koji obrađuje izlazni signal. Rade na način da odašiljač pošalje zraku svjetlosti. Najčešće se koriste infracrveno zračenje zato što je to svjetlost izvan vidljivog spektra, stoga je nevidljiva ljudskom oku, a i dioda se najmanje zagrijava. Prijemnik detektira to infracrveno svjetlo i šalje signal nalazi li se objekt u promatranom prostoru ili ne. Kao odašiljač koriste se svijetleće diode koje emitiraju infracrveno svjetlo, a kao prijemnik koristi se fototranzistor. Fototranzistor će detektirati svjetlost koja dolazi sa svijetleće diode tako što će kroz njega poteći struja dajući signal. Senzor može sadržavati svijetleću diodu i fototranzistor u istom kućištu ili oni mogu biti odvojeni. Kada su odvojeni, odašiljač i prijemnik se nalaze jedan nasuprot drugog, postavljeni tako da zraka svjetlosti upada točno na fototranzistor. Fototranzistor je aktivan sve dok objekt ne prođe kroz zraku što prekida infracrveno svjetlo i tada šalje signal o detekciji objekta. Kada su svijetleća dioda i fototranzistor u istom kućištu nasuprot njih može biti montirana reflektirajuća površina od koje se svjetlost odbija. Isto kao prije, prolaskom objekta prekida se svjetlost i senzor šalje signal o njemu. Ovisno o modelu, senzori mogu detektirati objekte na udaljenostima od nekoliko milimetara do više stotina metara. Koriste se kod robota za praćenje trake ili izbjegavanje prepreka, brojanje objekata, sigurnosnim sustavima i za druge primjene.

Korišteni senzor sastoji se od odašiljača (svijetleća dioda) koji šalje infracrveno svjetlo i prijemnika (fototranzistor). Nalaze se na istom modulu jedan pored drugog. Odašiljač konstantno šalje infracrvenu svjetlost, dok je za to vrijeme prijemnik neaktivan. Prolaskom objekta pored svijetleće diode, objekt će reflektirati svjetlost koja će doći do fototranzistora. Kada svjetlo pogodi bazu fototranzistora, to omogućuje struji da poteče kroz njega. Upravljački sklop na modulu detektira prolazak struje kroz fototranzistor šaljući signal o prolasku objekta do mikrokontrolera koji dalje koristi tu informaciju. Na modulu se nalazi i potencijometar čijim okretanjem povećavamo ili smanjujemo osjetljivost senzora što znači da mu treba manje reflektirajućeg infracrvenog svjetla kako bi protekla struja kroz fototranzistor. Naš senzor može na ovaj način mjeriti prolazak predmeta na udaljenosti između 2 i 30 cm. Senzor se sastoji od tri izlazna pina. Dva su za napajanje, VCC pin koristimo za snagu i možemo ga spojiti na 3.3 V ili 5 V pin na Arduinou, a GND na senzoru spajamo na GND pin na Arduinou. Treći pin senzora spajamo na neki od digitalnih pinova na Arduinou i on služi za komunikaciju s njim.



Slika 3.12. FC-51 infracrveni senzor spojen na Arduino Uno

3.6. Ostale komponente

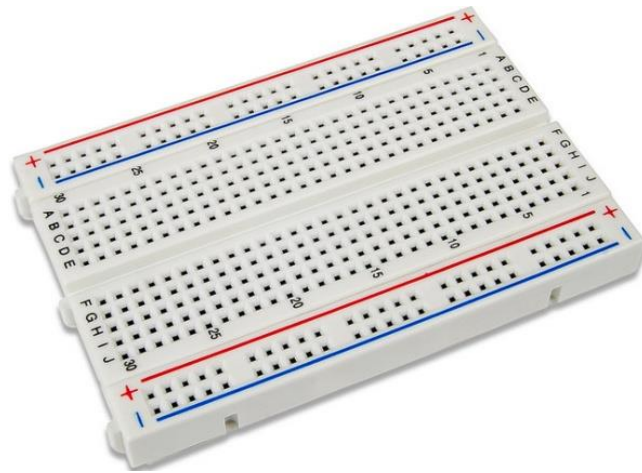
U prethodnom tekstu su opisani svi električni uređaji koje ćemo koristiti u našem projektu, ali moramo ih spojiti međusobno i sa izvorom napona.

3.6.1. Spojne žice

Električni vodič ne opire se prolasku struje kroz njega. Struja kroz vodič može stalno i kontinuirano teći uz gotovo zanemarivo opiranje tvari na njen prolazak (vrlo mali otpor). Žica je tako primjer električnog vodiča ako je napravljena od električki vodljivog materijala kao što su bakar ili aluminij. Kružnog je presjeka sa promjerom mnogo manjim od njezine duljine i električki je izolirana, osim na krajevima koji povezuju električne uređaje. Set žica koje su korištene napravljen je za povezivanje Arduino komponenata. Na svakom kraju žice nalaze se pinovi sa muškom ili ženskom konekcijom za jednostavno povezivanje. Korištene su žice duljine 30 cm i s tri tipa konekcije na pinovima, m/m, m/ž i ž/ž pinovi na krajevima žica. Za spajanje žice s Arduino koristi se muška konekcija, dok se za I2C modul koristi ženska konekcija.

3.6.2. Eksperimentalna pločica

Eksperimentalna pločica (eng. protoboard) predstavlja način za povezivanje električnih komponenata na ploči, kao što je vidljivo na slici 3.13. Na nju spajamo žice ili izravno komponente kao što su diode, otpornici ili tranzistori, ovisno o tome što želimo s čime povezati. Korisna je zato što ostvaruje kontakt između žica i komponenata bez potrebe za lemljenje i stvaranje trajnih spojeva i lako možemo mijenjati raspored spajanja. Na njoj lako možemo eksperimentirati sa spajanjem uređaja te razmotriti ideje spajanja dok ne dobijemo željeni spoj pa se koristi kao prototip za izradu električnih krugova. Kada smo zadovoljni rješenjem spajanja komponenata, možemo lako napraviti konačno rješenje spojeva u vidu tiskane pločice (PCB) koja predstavlja trajno rješenje. Eksperimentalna ploča sastoji se od mnogo utora koji ostvaruju kontakt sa žicom koja je spojena te je preko metalne trake povezuju sa drugim žicama u redu ili nizu. Ako gledamo pločicu na slici 3.13. po duljini, prva dva horizontalna reda na vrhu i na dnu pločice služe kao utori za napajanje i povezani su duž cijele linije. Između njih su međusobno spojene vertikalne linije uz prekid po sredini pločice. Prilikom spajanja, moramo paziti da ne spojimo komponente između utora koji su međusobno povezani da ne dođe do kratkog spoja zbog čega i imamo odvojene utore na pločici [16].



Slika 3.13. Eksperimentalna pločica

3.6.3. Napajanje

Napajanje je uređaj koji opskrbljuje električne uređaje električnom energijom. Ono povezuje izvor napona sa uređajem kojem je potreban. Može biti jednostavno kao kabel koji povezuje uređaje ili to mogu biti samostalni dijelovi opreme koji pretvaraju napon, struju i frekvenciju mreže ili nekog izvora na željenu vrijednost za predaju drugom uređaju. Sve korištene uređaje u projektu je potrebno napajati određenim naponom. Arduino Uno, kao najvažniji uređaj u ovom projektu, napajati ćemo na više načina. Arduino Uno ćemo prvo napajati preko USB kabela spojenog sa računalom. Preko njega će dobivati struju za rad i preko njega ćemo ga i programirati, odnosno prenijeti kod na njega. Kada želimo da Arduino bude lako prenosiv bez potrebe da bude priključen sa računalom povezati ćemo ga na zasebni izvor. Koristiti ćemo bateriju od 9 V. Biti će spojena na konekciju Arduina za napajanje. Taj napon je u rasponu između preporučenog napona od 7 do 12 V zbog čega će regulator napona na Arduinu smanjiti napon na radni napon od 5 V. Servomotore napajat ćemo iz posebnog izvora zato što prilikom rada koriste veliku struju koja ne bi bila dostupna da se napajaju sa Arduino pločice. Zato ćemo njih napajati iz posebnog izvora. Koristiti ćemo četiri AAA baterije. Svaka od njih ima napon od 1.5V. Tako ćemo spajanjem baterija dobiti napon od 6 V koji je dovoljan za napajanje servomotora. Ostale komponente napajati ćemo preko 5 V pina na Arduino pločici.

4. RAČUNARSTVO

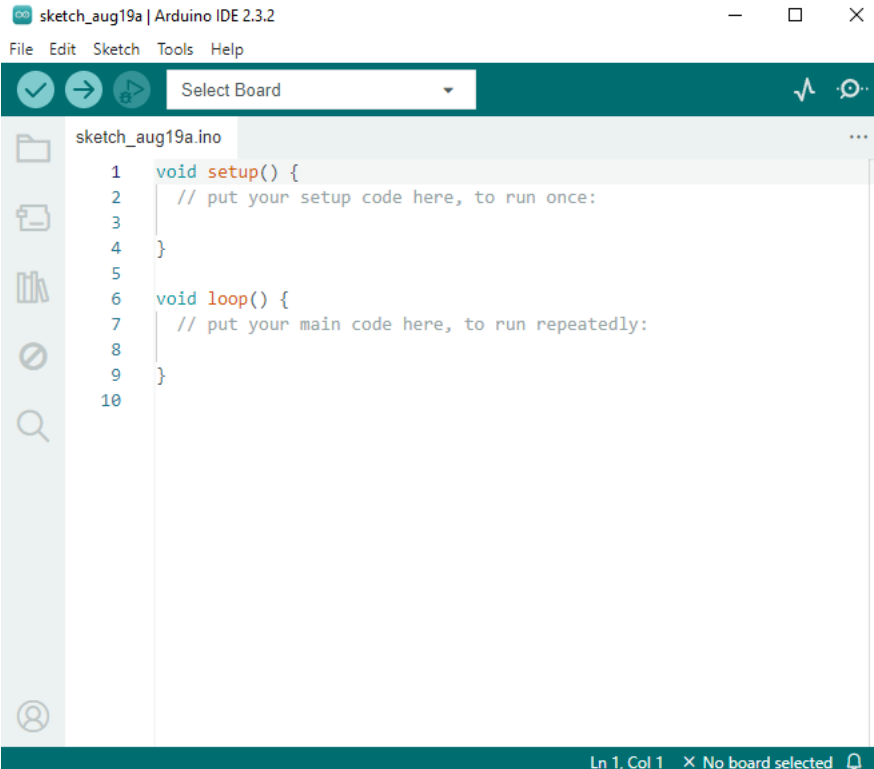
Računarstvo je znanost koja se bavi izgradnjom i uporabom sustava za računanje i obradu informacija. Obuhvaća teoriju, metode analize i sinteze, projektiranje i konstrukciju računalnih sustava. Programiranje je dio računarstva koje se bavi pisanjem koda, odnosno instrukcija računalu što i kako napraviti. Pomoću tog koda prenosimo naše želje i ideje računalu na način na koji ih može razumjeti. Programiranje se izvodi u nekom od programskih jezika u kojemu se pišu naredbe upotrebom određene sintakse i pravila koje su drugačije ovisno o programskom jeziku kojeg koristimo. Napisana pravila se tada prevode u strojni jezik pomoću programskog prevodioca (eng. compiler) kako bi ih računalo moglo razumjeti i izvršiti naredbe. Neki od programskih jezika su Python, C, BASIC, Java i drugi. Do sada smo se bavili sklopovljem, fizičkim dijelovima našeg projekta, a sada ćemo pogledati software koji uključuje sve nematerijalne dijelove projekta, kao što je programski kod koji izvršava hardware. U ovom slučaju napisani kod poslati ćemo mikrokontroleru na Arduino pločici koji će dalje upravljati komponentama. Vidjet ćemo i objasniti kako je napisan te proučiti njegove značajke [17].

4.1. Arduino IDE

Arduino IDE (eng. Integrated Development Environment) je programska podrška otvorenog tipa tvrtke Arduino za jednostavno programiranje Arduino pločica, a sučelje je prikazano na slici 4.1. To je programska podrška za Arduino u kojoj možemo napisati kod koji prebacujemo na pločicu kako bismo upravljali električnim uređajima. Koristi se za programiranje bilo kojeg tipa Arduino pločice. Podržava platforme Windows, Linux ili macOS i možemo ga besplatno preuzeti sa Arduino internetske stranice ili možemo raditi na mreži u Arduino Cloud Editor-u. Arduino IDE se aktivno razvija u tvrtci, a u projektu je korištena najnovija inačica Arduino IDE 2.3.2. Druge tvrtke mogu napraviti svoju verziju pločice sa drugačijim karakteristikama, pa je dobro što sve one mogu biti programirane od strane Arduino IDE programske podrške što značajno olakšava korištenje i razvoj ovakvih proizvoda. Isto tako, uz minimalna podešavanja programski kod će raditi i na drugim inačicama Arduino pločica. Programska podrška je napravljena na osnovi programskog jezika Java i C++ koji su u širokoj upotrebi. Uz dodatke jednostavnog sučelja koristi jednostavnu sintaksu i funkcije zbog čega je prilagođen za korištenje od strane početnika i onih koji nisu programirali, ali i profesionalaca. Sadrži uređivač teksta za pisanje koda, traku s alatima i niz izbornika. Zahvaljujući svestranosti, koristi se u velikom broju projekata. Sadrži brojne programske knjižnice (eng. library). Programska knjižnica sadrži zbirku podataka,

odnosno programski kod koji ne radi samostalno već se učitava u originalnom kodu kako bi nam omogućio dodatne funkcionalnosti. U njima se nalazi gotov kod sa nekom funkcijom koji lako učitamo, najčešće su to knjižnice s operacijama koje često koristimo. Korištene knjižnice definiramo na početku programskog koda i uključujemo naredbom `#include`. Programaska podrška dolazi sa zbirkom često korištenih knjižnica, a ostale možemo preuzeti s interneta. Upravo zbog velike i aktivne zajednice postoji veliki broj knjižnica koje su nam dostupne i mogu nam pomoći u rješavanju zadatka. Nakon što smo preuzeli željenu knjižnicu jednostavno ju uključimo u programu kako bi ju mogli koristiti [18].

Kod koji napišemo u programu zove se skica (eng. sketch). Kada pokrenemo Arduino IDE na računalu otvara se prozor sa početnom skicom koja se sastoji od dvije funkcije. Te dvije funkcije su najbitnije i moraju biti u svakom Arduino projektu. U `setup` funkciju upisujemo kod koji će biti pročitani i izvršeni jednom, a u `loop` funkciju upisujemo glavni kod koji će se izvršavati unedogled.



```
sketch_aug19a.ino
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
10
```

Ln 1, Col 1 × No board selected

Slika 4.1. Arduino IDE

Jednom kada završimo pisanje koda pritiskom na tipku verify provjerava se ima li napisana skica neke pogreške te ako ih nema kod je spreman za prebacivanje na pločicu pritiskom na tipku upload, a podaci će se slati preko USB kabela. Nakon što smo kod prebacili na Arduino pločicu

možemo reći da smo je programirali na način na koji želimo da radi. Jednom kada je pločica programirana ona, čim ju spojimo na napon, počinje sa izvršavanjem naredbi. Pritiskom na tipku za resetiranje pločica će iznova početi sa izvršavanjem koda. Pločica će tako iznova ponavljati radnje za koje smo je programirali i tek kada prebacimo drugačiji kod na nju ona će izvršavati te naredbe. Najvažniji koncepti koje ćemo koristiti prilikom programiranja su varijable i funkcije. Varijable su mjesta u koje spremamo informacije. Svaka varijabla ima svoje ime koje se odnosi na tip podatka koji u nju možemo spremiti. To mogu biti brojevi, slova ili skup slova, a svaki tip podatka ima posebni tip varijable. Nakon što smo odredili tip varijable, ovisno o tome što želimo spremiti u nju, možemo ju bilo kako nazvati i postaviti ju da bude jednaka željenoj vrijednosti. Funkcija je dio koda koji odrađuje definirani zadatak i znamo da se radi o funkciji kada iza riječi postoje otvorena i zatvorena zagrada. Kontrolna struktura predstavlja način na koji želimo da se programski kod izvršava korištenjem različitih funkcija [19].

4.2. Servomotor

Na slici 3.6. vidljivo je kako se može spojiti servomotor s Arduino pločicom. U nastavku je objašnjen programski kod koji je potreban za kontroliranje servomotora na jednostavnom primjeru koji ćemo kasnije koristiti prilikom pisanja glavnog koda.

```
#include <Servo.h>

Servo servomotor;

void setup() {
  servomotor.attach(9);
}

void loop() {
  servomotor.write(130);
  delay(5000);
  servomotor.write(90);
  delay(3000);
}
```

Prvo moramo uključiti knjižnicu za upravljanje servomotorima naziva Servo koja je unaprijed instalirana. Napravljena je nova varijabla tipa Servo i naziva servomotor kako bi se ubuduće u kodu moglo pozvati na nju. Zatim se ulazi u setup funkciju gdje se dodaje podatak da je korišteni servomotor priključen na pin 9 na Arduino pločici, a to se čini funkcijom attach. Funkcijom write upisuje se broj ovisno o brzini i smjeru u kojem želimo da se vrti korišteni servomotor, prema tablici 3.6. Izabran je broj 130 što znači da će se servomotor za to vrijeme okretati

brzinom između najmanje i najbrže. Funkcijom `delay` zaustavlja se izvršavanje koda na pet sekundi pa će se servomotor za to vrijeme okretati. Nakon toga se funkcijom `write` postavlja vrijednost 90 i servomotor će stati. Bit će zaustavljen tri sekunde prije nego što se ponovno počne izvršavati funkcija `loop`.

4.3. LCD zaslon

Na slici 3.9. vidljivo je kako spojiti zaslon s Arduino pločicom. Kada se koristi I2C modul, prvo moramo pronaći njegovu adresu kako bi Arduino mogao s njim komunicirati. Kako bi pronašli adresu, koristi se programski kod koji je preuzet s interneta i koji služi specifično za traženje te adrese. Nakon preuzimanja koda otvaramo ga u Arduino IDE-u, gdje ćemo ga prebaciti na pločicu. Arduino će tada naći adresu koju se čita otvaranjem serial monitora u programu. Dobivena adresa je 0x27, a kod za upravljanje zaslonom je naveden u nastavku teksta.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  lcd.init();
  lcd.backlight();
}

void loop() {
  lcd.setCursor(4, 0);
  lcd.print("Tehnicki");
  lcd.setCursor(4, 1);
  lcd.print("fakultet");
  delay(5000);
  lcd.clear();
}
```

Prvo se učitava knjižnica `Wire` koja je također predinstalirana. Ona omogućuje korištenje I2C komunikacije između Arduina i uređaja. `LiquidCrystal` knjižnica već je dostupna prilikom instalacije programa, ali budući da se zaslon kontrolira preko I2C modula uključena je `LiquidCrystal_I2C` knjižnica koja se koristi u tu svrhu i preuzeta je s Arduino internetske stranice. Nakon toga, definira se varijabla tipa `LiquidCrystal_I2C` koja je nazvana `lcd` i u koju se sprema podatke o korištenom zaslonu. Prvo se navodi prethodno pronađena adresa zaslona i zatim se upisuje broj redova i broj stupaca zaslona koji se koriste odnosno 16 i 2. U `setup` funkciji inicijalizira se `lcd` i uključuje pozadinsko osvjetljenje. U `loop` funkciji upravlja se željenim ispisom na zaslonu načinom koji je opisan u nastavku teksta. Funkcijom `setCursor`

određuje se gdje želimo početi pisati riječ na zaslonu. Sastoji se od dva broja, a prvi govori u kojem stupcu će biti prvi znak riječi (od 0 do 15 u ovom slučaju). Drugi broj govori u kojem redu će biti prvi znak riječi (0 ili 1 u ovom slučaju). Treba imat na umu da se riječi neće same početi ispisivati u drugom redu ako u prvom više nema mjesta. Funkcijom `print`, počevši od četvrtog stupca, ispisuje se riječ Tehnicki u sredini zaslona, postavlja se pokazivač u drugi red i ispisuje riječ fakultet. Na zaslonu će riječi biti prikazane pet sekundi nakon čega će funkcijom `clear` sa zaslona nestati bilo kakvi znakovi.

4.4. Tipkovnica

Tipkovnicu se povezuje tako da su pinovi redom povezani na digitalne pinove Arduina od 3 do 9. Kod za upravljanje tipkovnicom je naveden u nastavku teksta.

```
#include <Keypad.h>

const byte Red = 4;
const byte Stupac = 3;

char tipke[Red][Stupac] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};

byte pinovired[Red] = {3,4,5,6};
byte pinovistupac[Stupac] = {7,8,9};
Keypad tipkovnica = Keypad(makeKeymap(tipke), pinovired, pinovistupac, Red, Stupac);

void setup() {
  Serial.begin(9600);
}
void loop() {
  char pritisnutatipka = tipkovnica.getKey();
  Serial.println(pritisnutatipka);
}
```

Uključena je knjižnica `Keypad` koja je preuzeta s Arduino internetske stanice. Definirane su dvije varijable tipa `const byte` u koje se spremaju podatci o korištenoj tipkovnici koji se ne mijenjaju. Jedna varijabla se naziva `Red`, a druga `Stupac` te su postavljene tako da budu jednake broju redova i stupaca. Nakon toga definirana je varijabla tipa `char` koja služi za spremanje znakova, nazvana je `tipke` i određene su joj granice stavljajući u uglate zagrade red i stupac. Između vitičastih zagrada zadaje se niz (eng. `array`) koji predstavlja strukturu podataka jednake veličine

memorije da su podaci zasebno spremljeni, a svaki podatak dostupan je pomoću posebnog ključa. Sastoji od 4 reda i 3 stupca u koje se zapisuju znakovi tipkovnice koji se pohranjuju u varijablu char. Nakon toga, definiraju se pinovi na koje je korištena tipkovnica povezana. Redovi tipkovnice povezani su na pinove 3, 4, 5 i 6, a stupci na pinove 7, 8 i 9. Konstruiran je objekt tipkovnica koji definira korištenu tipkovnicu koristeći funkcije Keypad i makeKeymap, a u njega će se pohraniti sve prethodno konstruirane varijable. Da bi se omogućio unos pritisnute tipke sa tipkovnice u programski kod, definira se još jedna varijabla tipa char u koju se sprema vrijednost ovisno o tome koju se tipku pritisne. Varijabla se naziva pritisnutatipka i jednaka je funkciji getKey koja čita pritisnuti znak i sprema ga u varijablu. Kako bi bili sigurni da kod radi ispravno pomoću funkcija Serial.begin i Serial.println unutar programa ispisujemo pritisnutu tipku na monitoru.

4.5. Senzor blizine

Prema slici 3.12. spaja se senzor blizine. Kod za senzor blizine slijedi u nastavku teksta.

```
void setup() {
  pinMode(9, INPUT);
  Serial.begin(9600);
}

void loop() {
  if(digitalRead(9)==LOW) {
    Serial.println("ima predmeta");
  }
}
```

Za senzor blizine nije potrebno koristiti knjižnice. Pomoću funkcije pinMode definira se na koji pin na Arduino pločici je spojen signal sa senzora i definira se da se radi o ulazu. Zato će se u ovom slučaju digitalni pin ponašati kao ulaz. Isto kao i u prethodno primjeru, uključen je serial monitor da bi se promatralo njegovo ponašanje. Funkcija if dio je kontrolne strukture u kojem se provjerava je li navedena tvrdnja istinita. Ako je istinita, dio koda između vitičastih zagrada nakon funkcije izvršit će se, a ako nije istinita taj dio koda se zanemaruje u potpunosti. Na taj način pomoću funkcije if uspoređujemo vrijednosti, što se označava dvostrukim znakom jednakosti. Provjerava se je li ulaz u pin 9 jednak vrijednosti HIGH ili LOW. Kada senzor ne očitava nikakav predmet, ulaz na pinu 9 bit će u stanju HIGH, što znači da će se kod koji slijedi zanemariti, a kada senzor očita prolazak predmeta ulaz u pin 9 bit će u stanju LOW zbog čega će uvjet postati istinit i izvršiti će se daljnji kod koji će na monitoru ispisati ima predmeta.

5. STROJARSTVO

Strojarstvo je znanost koja se bavi projektiranjem, proizvodnjom, analizom i održavanjem mehaničkih sustava i predmeta. Bavi se proučavanjem prirodnih znanosti kao što su fizika i matematika zajedno s raznim znanostima kao što su znanost o materijalima, dinamika, mehanika i druge znanosti. Strojarstvo djeluje kako bi konstruirali i razvili neki novi proizvod, stroj ili dio stroja koristeći razne tehnike i alate.

Za konačni dio projekta važna je vanjska konstrukcija, odnosno kućište u koje se smještaju sve električne komponente da bi mogle obavljati svoju funkciju. Kućište se odnosi na vanjski izgled i funkcionalnost cijelog projekta. Moramo biti konstruirano tako da zadovoljava funkciju samoposlužnog automata. Za materijal kućišta u ovom projektu korišteno je drvo. Drvo je korišteno zbog jednostavnosti obrade, pristupačnosti, niske cijene, velikog izbora vrsta i veličina te posjedovanja alata za njegovu obradu. Kako je kućište drveno, a drvo nije klasični materijal korišten u strojarstvu, za razliku od metala, preciznije je reći da se radi o stolarstvu. Stolarstvo je djelatnost koja se bavi izradom predmeta i konstrukcija od drva.

5.1. Materijali i strojevi

Za izradu kućišta korišteni su sljedeći materijali. Ploče od drva koje ćemo koristiti su iverica debljine 10 mm, kantirana iverica debljine 22 mm i lesomit debljine 5 mm. Iverica je svaka drvena ploča nastala lijepljenjem ivera (piljevine, malih i tankih listića) drva ili nekih drugih prirodnih materijala (slama, konoplja) pod pritiskom sa sintetskim ili prirodnim ljepilima uz djelovanje tlaka i topline. Kantirana iverica predstavlja ivericu na kojoj su sa svake strane zalijepljene trake iz estetskih razloga i zbog očuvanja mehaničkih svojstva ploče. Lesomit se sastoji od međusobno isprepletenih vlakana drva proizvedenih mokrim postupkom zagrijavanjem uz dodatak vezivnih sredstva [20].

Svaku od tih ploča potrebno je izrezati prema željenim dimenzijama. Strojevi koji su korišteni za obradu drva su ubodna pila, prikazana na slici 5.1, bušilica i višenamjenska brusilica. Alati koji su korišteni su turpije i odvijači. Ostali predmeti koji su korišteni su vide za drvo različitih veličina, kruna za drvo, željezna žica, matice, groverice, šajbice, šarniri, kutnici, pleksiglas i magnet.



Slika 5.1. Korištena ubodna pila

6. IZRADA SAMOPOSLUŽNOG AUTOMATA

U prethodnim poglavljima su objašnjeni svi principi, uređaji i metode, sklopovlje i programska podrška koji su korišteni za izradu samoposlužnog automata. U ovom poglavlju opisuje se rad svih prethodno spomenutih dijelova povezanih u smislenu cjelinu. Bit će objašnjen način povezivanja električnih komponenata u jednu cjelinu i pisanje koda koji pomoću kojeg će Arduino Uno upravljati električnim elementima. Nakon toga detaljno je opisana izrada kućišta i ugradnja ostalih komponenata u njega. Biti će objašnjen princip rada gotovog projekta te njegova svrha i primjena. Samoposlužni automat čija izrada je opisana je pojednostavljena verzija onih automata koje možemo vidjeti u svakodnevnicima, ali dijeli njihovu svrhu kao i sve glavne dijelove.

6.1. Povezivanje električnih elemenata

Prije ovog ulomka je naveden i objašnjen način rada za sve pojedinačne električne komponente koje su korištene, a u ovom opisano je kako se spajaju u jednu cjelinu koja čini električni dio samoposlužnog automata. Komponente su Arduino, dva servomotora, LCD zaslon, infracrveni senzor blizine, 3x4 tipkovnica, spojne žice, eksperimentalna pločica i napajanje od 6 i 9 V.

Arduino je središte projekta i na njega se izravno povezuju signalni kabeli. Korištenjem sedam m/m žica 3x4 tipkovnica je izravno povezana s Arduinoom. Svaki od pinova tipkovnice spojen je sa jednom žicom. Drugi kraj žice spojen je na digitalne pinove na Arduinou na sljedeći način. Prve četiri žice koje predstavljaju stupce spojene su na pinove 3, 4, 5 i 6. Ostale tri žice koje predstavljaju stupce spojene su na pinove 7, 8 i 9. Tipkovnica je nakon ovih radnji spojena i spremna za rad. Zatim se spaja LCD zaslon. Pinovi I2C modula koji je već povezan na zaslon povezuju se na sljedeći način. Korištenjem m/ž žica povezuje se SDA pin sa modulu na SDA pin na Arduinou i SCL pin na modulu povezuje se SCL pinom Arduinoa. IR senzor blizine povezati ćemo m/ž žicom tako što pin senzora OUT povezuje na pin 2 na Arduino pločici. Servomotori nemaju označene pinove, ali narančasti pin označava signal za upravljanje servomotorom. Pin jednog od servomotora spaja se preko m/m kabela na pin 10 Arduinoa, a pin drugog servomotora na pin 11 na Arduino pločici. Ovime su nabrojani svi pinovi elemenata koji se izravno spajaju na Arduino.

U nastavku teksta opisano je spajanje pinova snage, odnosno pinova pomoću kojih komponente dobivaju električnu energiju. Za povezivanje pinova snage koristi se eksperimentalna pločica. Na

dvije horizontalne linije povezuje se napajanje od 6 V tako da se u jednu liniju povezuje minus pol napajanja, a u drugu plus pol napajanja. Preko tog napajanja napaja se servomotor. Od preostala dva pina na svakom od servomotora, onaj crvene boje predstavlja plus pol i m/m žicom se povezuje na liniju u koju je spojen plus pol napajanja (također crvena žica), a žicu smeđe boje koja predstavlja minus pol spaja se u liniju u koju je spojen minus pol napajanja (žica crne boje). Na ovaj način servomotori će primati snagu za rad isključivo preko napajanja od 6 V. IR senzor blizine i LCD zaslon napajaju se preko Arduino pločice. Za to se koristi pin snage od 5 V. Jednom m/m žicom povezuje se pin 5 V Arduina na jedan od vertikalnih pinova eksperimentalne pločice, a u istom redu veza na eksperimentalnoj pločici povezuje se VCC pin senzora blizine i VCC pin LCD zaslona. Na taj način se u istu točku spaja pin Arduina 5 V i pinovi za napajanje senzora i zaslona koji se tako napajaju. Pinovi oznaka GND senzora blizine i zaslona koji predstavljaju minus pol spajaju se u istu vertikalnu liniju eksperimentalne pločice u koju su već spojeni minus polovi servomotora i minus pol napajanja.

Preostao je najvažniji korak u kojem se jednom žicom povezuje sve negativne polove elemenata u zajedničku točku kako bi koristili istu referentnu točku, odnosno uzemljenje. To se čini korištenjem jedne žice kojoj se jedan kraj spaja u liniju gdje su spojeni svi negativni polovi komponenata, a drugi kraj spaja se na GND pin na Arduino pločici i time dobiti zajedničko uzemljenje. Dovršetkom ovog koraka, spojene su sve žice svih komponenata električne sheme. Drugo napajanje od 9 V spaja se izravno na utor Arduina za njegovo napajanje, ali u kasnijim fazama izrade.

Zašto se servomotori napajaju preko posebnog napajanja, a ne preko istog pina snage od 5 V preko kojeg se napajaju senzor i zaslon? Servomotori su komponente koje u opisanom primjeru zahtijevaju veliku struju za svoj rad koju im Arduino, bez dodatnog napajanja, ne bi mogao isporučiti. Ako se Arduino napaja preko USB veze maksimalna struja je ograničena na 500 mA, a to je struja koju će Arduino moći isporučiti komponentama. Ako se komponente napajaju preko 5 V pina, mora se uzeti u obzir da će taj broj biti u početku manji zato što Arduino koristi oko 50 mA za svoj rad. Ako su korišteni još neki od izlazno/ulaznih pinova, a oni koriste oko 20 mA struje. U slučaju napajanja komponenata koje koriste struju veću od one koju Arduino može isporučiti, doći će do neprestanog ponovnog pokretanja Arduina zbog nestanka struje, a može doći i do oštećenja. Ako se Arduino napaja preko posebnog ulaza za napajanje, Arduino može isporučiti struju od 1 A, a to još uvijek nije dovoljno. Ta bi struja bila dostatna najviše za jedan slabiji servomotor bez tereta na njemu. Kako se koriste dva jača servomotora, od kojih svaki koristi struju od 170 mA bez tereta, a do 1.2 A uz maksimalan teret, mora ih se napajati preko

posebnog napajanja koje omogućuje veću potrošnju struje i zbog toga se koriste baterije. LCD zaslon troši oko 70 mA za rad, a senzor blizine oko 40 mA, a to je dovoljno mala struja da ih se napaja preko pina od 5 V na Arduino pločici.

6.2. Programiranje Arduino Uno pločice

Nakon što su sve komponente električki povezane, potrebno je napisati kod pomoću kojeg će Arduino Uno njima upravljati. Ovo je glavni kod koji služi za upravljanje samoposlužnim automatom. Napisani glavni programski kod je sljedeći.

```
#include <Wire.h>
#include <Keypad.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

Servo servo1, servo2;

const byte Red = 4;
const byte Stupac = 3;

char tipke[Red][Stupac] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};

byte pinovired[Red] = {3,4,5,6};
byte pinovistupac[Stupac] = {7,8,9};

Keypad tipkovnica = Keypad(makeKeymap(tipke), pinovired, pinovistupac, Red, Stupac);

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {

  lcd.init();
  lcd.backlight();

  servo1.attach(10);
  servo2.attach(11);

  pinMode(2, INPUT);
}

void loop() {

  lcd.clear();
  lcd.setCursor(4, 0);
```

```

lcd.print("Ubacite");
lcd.setCursor(4, 1);
lcd.print("kovanicu");

while (true) {
    if(digitalRead(2) == LOW) {
        break;
    }
}
while (true){
char pritisnutatipka = tipkovnica.getKey();

lcd.setCursor(4, 0);
lcd.print("Izaberite");
lcd.setCursor(4, 1);
lcd.print("proizvod");
delay(50);

if (pritisnutatipka=='1'){
    lcd.clear();
    lcd.setCursor(2, 0);
    lcd.print("Odabrali ste");
    lcd.setCursor(3, 1);
    lcd.print("proizvod 1");
    delay(2000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Dostavljanje...");
    servo1.write(130);
    delay(1940);
    servo1.write(90);
    delay(1000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Hvala na kupnji!");
    delay(3000);
    break;
}

if (pritisnutatipka=='9'){
    lcd.clear();
    lcd.setCursor(2, 0);
    lcd.print("Odabrali ste");
    lcd.setCursor(3, 1);
    lcd.print("proizvod 9");
    delay(2000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Dostavljanje...");
    servo2.write(130);
    delay(1705);
    servo2.write(90);
    delay(1000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Hvala na kupnji!");
}
}

```

```

    delay(3000);
    break;
}

if (pritisnutatipka == '2' || pritisnutatipka == '3' || pritisnutatipka == '4'
    || pritisnutatipka == '5' || pritisnutatipka == '6' || pritisnutatipka == '7'
    || pritisnutatipka == '8' || pritisnutatipka == '*' || pritisnutatipka == '0'
    || pritisnutatipka == '#') {

    lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("Nepostojeci");
    lcd.setCursor(3, 1);
    lcd.print("proizvod");
    delay(1300);
    lcd.clear();
}
}
}

```

Prvo su uključene četiri potrebne knjižnice. Definirana su dva servomotora koja su nazvani servo1 i servo2. Napravljen je objekt tipkovnica i zatim je definirana adresa te broj redova i stupaca LCD zaslona. Nakon toga se ulazi u setup funkciju u kojoj se uključuje pozadinsko osvjetljenje zaslona, definira se da je signal sa servo1 servomotora spojen na pin 10, a signal sa servo2 servomotora spojen na pin 11. Senzor blizine povezan je s pinom 2 koji se ponaša kao ulaz. Nakon toga, ulazi se u loop funkciju.

Na zaslonu se prvo ispisuje tekst „Ubacite kovanicu“, a nakon toga se ulazi u while petlju. While petlja je kontrolna struktura čiji će rad biti opisan u idućim rečenicama. Sve dok je uvjet koji smo napisali u obliku zagrada while petlje točan, izvršava se samo dio koda između vitičastih zagrada vezan za while petlju, a ako uvjet više nije točan izlazi se iz petlje i program nastavlja dalje čitati kod. Unutar while petlje uključena je funkcija if koja se neprestano provjerava jer se nalazi u while petlji. Kada senzor detektira prolazak predmeta stanje u if funkciji postaje točno nakon čega se izvršava kod unutar if funkcije koji sadrži naredbu break. Break je također kontrolna struktura pomoću koje izlazimo iz petlje neovisno o tome je li petlja zadovoljila uvjet. Time se izlazi iz prve while petlje i ulazi se u sljedeću.

U drugoj while petlji izvršavaju se naredbe opisane u nekoliko idućih rečenica. Prvo se poziva funkcija getKey koja, u slučaju da je određena tipka na tipkovnici pritisnuta, odgovarajući znak sprema u char varijablu koja je nazvana pritisnutatipka. Ako se neka tipka tipkovnice pritisne prije nego što se pozove ova funkcija, ništa se ne događa. Zatim se na zaslonu ispisuje tekst „Izaberite proizvod“ nakon čega dolazimo do tri if funkcije. Kako bi se izvršio kod u nekoj od

navedenih triju funkcija, mora biti pritisnuta neka od tipki na tipkovnici. Sve dok niti jedna tipka nije pritisnuta, izvršava se samo dio koda unutar while petlje. Drugim riječima, program čeka da se izabere neka tipka na tipkovnici.

Ako se pritisne tipka 1, uvjet prve if funkcije bit će točan i izvršava se kod vezan za nju. Na zaslonu se u tom slučaju prikazuje tekst „Odabrali ste proizvod 1“ koji će biti na zaslonu dvije sekunde, zatim se na zaslonu ispisuje „Dostavljanje...“. Dolazi do okretanja jednog od servomotora koje traje 1940 ms nakon čega motor staje. Jednu sekundu nakon zaustavljanja motora na zaslonu se ispisuje „Hvala na kupnji !“, a prikaz tog teksta traje tri sekunde. Ako umjesto tipke 1 pritisnemo 9 uvjet druge if funkcije bit će točan i izvršava se kod vezan za nju. Na zaslonu se u tom slučaju prikazuje tekst „Odabrali ste proizvod 9“ koji će biti na zaslonu dvije sekunde, zatim se na zaslonu ispisuje „Dostavljanje...“. Dolazi do okretanja drugog servomotora koje traje 1705 ms nakon čega motor staje. Jednu sekundu nakon zaustavljanja motora na zaslonu se ispisuje „Hvala na kupnji !“ koje traje tri sekunde. Nakon toga naredbom break izlazimo iz druge while petlje koja je ujedno i posljednja stavka u zapisanom kodu. Nakon toga se program vraća na početak glavne petlje loop i cijeli kod se izvršava ponovno.

Ako pritisnemo bilo koju tipku tipkovnice osim tipki 1 i 9, uvjet treće if funkcije bit će točan. Znak || u programskom jeziku znači „ili“, a to znači da će se, ukoliko je bilo koji od uvjeta točan, izvršiti kod vezan za treću if funkciju. Na zaslonu će se tada ispisati „Nepostojeci proizvod“, a to traje 1300 ms. Nakon treće if funkcije nema naredbe break kao u prethodna dva slučaja pa se program vraća na izvršavanje druge while petlje. Iz druge while petlje moguće je izaći samo kada odaberemo tipku 1 ili 9.

6.3. Izrada kućišta

U ovom ulomku opisan je proces izrade kućišta koristeći navedene materijale i strojeve. Kako se izrađuje prototip samoposlužnog automata, mora biti jednostavan, ali ujedno održavati njegovu glavnu funkciju. Osnovne značajke automata su odvojeni prostor za smještaj proizvoda koje želimo prodavati, mjesto gdje će proizvod doći da ga osoba može preuzeti i prostor za smještaj svih električnih komponenti. Utor za kovanice, tipkovnica i zaslon moraju biti na vidljivom mjestu. Imajući sve u vidu, prvo su razrađene dimenzije samoposlužnog automata tako da u njemu bude dovoljno prostora za smislen smještaj navedenih elemenata. Korištenjem ubodne pile izrezane su navedene drvene ploče željenih dimenzija. U tablici 6.1 dostupan je prikaz svih ploča koje su korištene.

Tablica 6.1. Dimenzije izrezanih ploča

Vrsta ploče:	Dimenzije:	Opis:
Kantirana iverica1	370x290 mm	Donja strana
Kantirana iverica2, komada 2	290x350 mm	Bočna strana
Kantirana iverica3	370x185mm	Središnja strana
Iverica1, komada 3	185x150 mm	Središnja strana
Iverica2	210x170 mm	Unutarnja stražnja strana
Lesonit1	70x330 mm,	Unutarnja bočna strana
Lesonit2	215x330 mm	Unutarnja prednja strana
Iverica3	345x410 mm	Prednja strana
Iverica4	415x338 mm	Gornja strana
Iverica5	410x340 mm	Stražnja strana

Neke od navedenih ploča zahtijevaju dodatno rezanje unutar njih. Za izradu utora unutar drva korištene su krune za bušenje drva koje se montiraju na bušilicu. Pomoću krune izrađena je rupa u svakom kutu željenog otvora nakon čega je ubodnom pilom izrezano drvo između njih kako bi se izradio otvor. Rubove i neravnine napravljenog otvora izgladene su brušenjem koristeći višenamjensku brusilicu i/ili turpiju.

Unutar komponente iverica2 potrebno je napraviti rupe u koje se smještaju servomotori. Za svaki motor potrebno je napraviti poseban utor koristeći objašnjeni postupak. Dimenzije motora su 21 x 41 mm pa se izrađuju rupe tih dimenzija uz dodatak na dnu za prolazak kabela. Ovdje je potrebno biti precizan tako da rupe budu pozicionirane točno između utora u koje se stavljaju proizvodi.



Slika 6.1. Iverica2 s utorima za motore

Iverica3 je prednja strana automata i zato je na njoj potrebno napraviti utor za pristup predmetu prilikom preuzimanja, prozor kako bi se predmeti mogli vidjeti te utor za kovanicu, zaslon i tipkovnicu. Dimenzije su redom sljedeće 175 x 75 mm, 175 x 132 mm, 4 x 28 mm, 75 x 35 mm, 25 x 10 mm. U lesonit2 potrebno je napraviti rupu veličine 165 x 125 mm.

Izradu počinjemo ugradnjom bočnih strana (kantirana iverica2) na postolje (kantirana iverica1). Prije nego se između bočnih strana ugradi središnju ploču (kantirana iverica3), na nju moramo ugraditi tri komada (iverica1) koji predstavljaju strane između kojih se stavljaju proizvodi. Za povezivanje kantiranih iverica koriste se vide za drvo 4 x 45 mm, a za povezivanje kantirane iverice sa ivericom vide 3 x 30 mm. Sve vide zavidavaju se odvijačem.



Slika 6.2. Dio izrade kućišta

Nakon toga ugrađuje se unutrašnjost kućišta. Servomotere ugrađuje se u ivericu2 i ona se postavlja sa stražnje strane, a sa prednje strane lesonit2. Za ugradnju se koriste vide 3 x 20 mm. Sprijeda, pod pravim kutom na lesonit2, se postavlja lesonit1 i pričvršćuje se koristeći dva kutnika. Nakon toga prelazi se na vanjske strane. Kao stražnju stranu je korištena iverica5 koja se ugrađuje koristeći vide 3.5 x 30 mm. Prednju stranu (iverica3) ugrađuje se koristeći šarnire na desnom rubu kućišta kako bi se ta strana mogla otvoriti, a na lijevi rub postavlja se magnet kako

bi prednja strana ostala zatvorena prilikom korištenja. Također uz pomoć šarnira, ugrađuje se gornju stranu (iverica4) na stražnju kako bi se i gornja strana mogla otvoriti. Za ugradnju šarnira korištene su vide 3 x 10 mm. Otvor koji služi da se mogu vidjeti dostupni proizvodi prekriven je pleksiglasom. Na unutrašnjost prednje strane, na dijelu kroz koji prolazi ubačena kovanica postavlja se komad lima tako da ubačena kovanica ona padne ravno dolje kako bi sigurno aktivirala senzor. Ovim korakom završava izgradnja kućišta. Unutar se dodaje električne elemente.

Kako bi se predmeti izbacivali jedan po jedan, izrađena je žičana spirala. Napravljena je tako što je komad željezne žice debljine 1.8 mm namotan oko boce spreja promjera 50 mm. Žica je namotana da se dobije oblik spirale. Zbog krutosti žice spirala je promjera 65 mm, što odgovara primjeni. Na dio plastike koji se stavlja na servomotor nije moguće pričvrstiti napravljenu spiralu pa je metričkom vidom i maticama napravljena distanca između djela koji se stavlja na motor i okruglog željeznog čepa. U sredini čepa izrađena je rupa kroz koju je provučena žica napravljene spirale i osigurana je tako da se ne bi mogla pomaknuti. Tijekom izrade potrebno je paziti da razmak između svakog luka na spirali bude jednak i dovoljno širok za proizvod koji želimo staviti unutar nje. Prva spirala ima prostora za 5 predmeta i nju se spaja sa servomotorom koji je ugrađen lijevo u kućištu, a prilikom pisanja programa nazvan je servo1. Druga spirala ima prostor za 4 predmeta i spaja se na servomotor ugrađen desno i u kodu smo nazvan servo2.



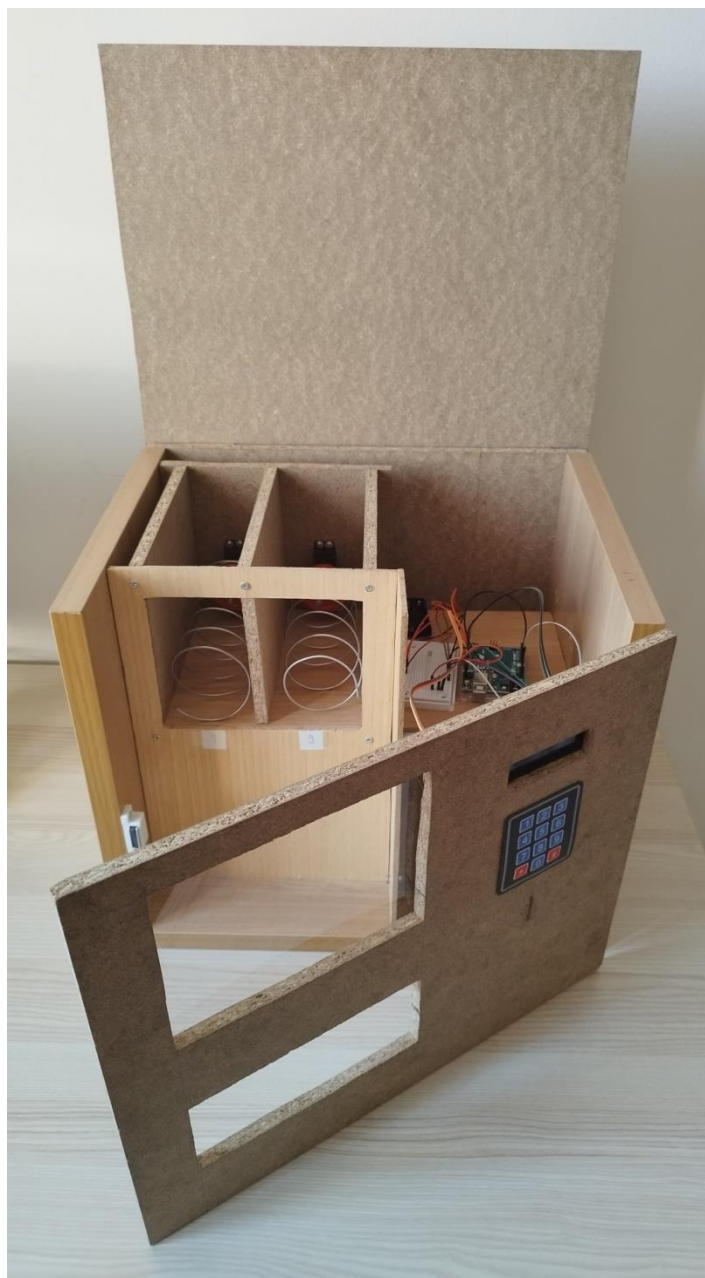
Slika 6.3. Žičane spirale

U objašnjenju glavnog koda koji Arduino koristi za upravljanje svim komponentama, spominje se vrijeme tijekom kojeg se neki od motora vrte. Za servo1 vrijeme iznosi 1940 ms, što znači da je toliko vremena potrebno kako bi spirala napravila jedan puni okretaj, a za servo2 vrijeme iznosi 1705 ms. Vremena nisu jednaka jer je korišteno manje žice prilikom izrade druge spirale za 4 predmeta pa se taj servomotor kraće mora okretati da se spirala vrati u početni položaj. Vrijeme potrebno za jedan okretaj spirale dobiva se testiranjem različitih vremena sve dok se ne pronađe traženo vrijeme za svaki motor zasebno. To je posljednji korak izrade prototipa samoposlužnog automata.



Slika 6.4. Prototip samoposlužnog automata temeljen na Arduinu

Neke od karakteristika automata su navedene u idućim rečenicama. Dimenzije su mu 415 x 360 x 338 mm, teži 11.6 kg, sastoji se od dva odvojena prostora u koje je moguće staviti sveukupno 9 proizvoda. Dimenzije proizvoda koji se stavlja u prostor ne bi smjele biti veće od 75 x 125 mm, lako se nadopunjuje otvaranjem gornje strane, a kovanice se uzimaju otvaranjem prednje strane.



Slika 6.5. Prikaz otvorenih strana samoposlužnog automata

6.4. Princip rada

Prototip samoposlužnog automata funkcioniра prema principu koji je opisan u ovom ulomku. Osoba koja dođe do njega lako može vidjeti koji proizvodi se nalaze u njemu. Ako je osoba zainteresirana za kupnju jednog od proizvoda na ekranu može pročitati „Ubacite kovanicu“. Nakon što je kovanica ubačena ona prolazi pored senzora blizine koji je detektira i daje signal o ubačenoj kovanici. Na zaslonu se tada ispisuje „Izaberite proizvod“. Ispod proizvoda s lijeve strane nalazi se broj 1, a ispod onog s desne strane broj 9. Osoba tada pomoću tipkovnice odabire broj ovisno o proizvodu koji želi kupiti. Ako osoba pritisne neki broj koji nije jedan od

ponuđenih na zaslonu će se ispisati „Nepostojeci proizvod“ i osoba će opet biti u mogućnosti izabrati jedan od dva proizvoda. Ako osoba pritisne broj 1 na zaslonu će se ispisati „Izabrali ste proizvod 1“. Ubrzo nakon ispisuje se „Dostavljanje...“ i servomotor koji okreće žičanu spiralu tog proizvoda će se okrenuti zbog čega će jedan primjerak tog tipa proizvoda pasti u prostor gdje ga osoba može jednostavno preuzeti. Osoba tada treba preuzeti proizvod, a na zaslonu će pisati „Hvala na kupnji!“. Isto bi se dogodilo i da je osoba izabrala proizvod pod brojem 9, ali u tom bi se slučaju okrenula spirala vezana za taj proizvod. Na zaslonu se tada ispisuje „Ubacite kovanicu“ i cijeli proces počinje ispočetka. Kada želimo nadopuniti automat proizvodima ili zamijeniti baterije otvaramo njegovu gornju stranu i proizvode stavljamo između lukova spirale. Ukoliko želimo preuzeti kovanice otvaramo prednju stranu kako bismo ih uzeli.

7. ZAKLJUČAK

U ovom radu izrađen je Prototip samoposlužnog automata. Cilj projekta je napraviti uređaj pomoću kojeg osoba lako može kupiti neki proizvod samostalno bez potrebe za drugom osobom koja bi bila prodavač tog proizvoda ili nadgledala prodaju. Projekt se temelji na Arduino Uno pločici kao glavnoj komponenti pomoću koje se upravlja ostalim komponentama. Kako bi se izradio projekt, korištena su znanja iz više tehničkih znanosti koja su objedinjena kako bi se izradio željeni uređaj. Razmotrene su potrebne električke komponente, proučen je i objašnjen njihov rad i osmišljena je električna shema njihovog spajanja. Projekt se bavi programiranjem pomoću Arduino IDE programa u kojem je napisan programski kod koji pločica koristi za izvršavanje danih naredbi. Kućište je napravljeno od drva i sadrži prostor za proizvode, dio za preuzimanje proizvoda te sučelje koje korisnik koristi prilikom kupovine. Sučelje se sastoji od zaslona, tipkovnice i utora za kovanice koje po ubacivanju prolaze pored senzor blizine. Unutrašnjost automata jednostavno je dostupna pomoću otvaranja prednje i gornje stranice. Korisnik može ubaciti kovanicu i odabrati proizvod koji uređaj ubacuje u prostor gdje ga osoba može preuzeti, a na taj način je automatiziran proces prodaje.

LITERATURA

- [1] Wikipedia: „Vending machine“, s interneta, https://en.wikipedia.org/wiki/Vending_machine, kolovoz 2024.
- [2] Wikipedia: „Automatika“, s interneta, <https://hr.wikipedia.org/wiki/Automatika>, kolovoz 2024.
- [3] Wikipedia : „Arduino“, s interneta, <https://en.wikipedia.org/wiki/Arduino>, kolovoz 2024.
- [4] Arduino.cc: „Arduino Uno Rev3 SMD“,s interneta, <https://store.arduino.cc/products/arduino-uno-rev3-smd?srsId=AfmBOopG9fXyjVoCj4ZLyvEPd6KerWtOvenYnJ0m5zBkjsOhPf2HUGrY>, kolovoz 2024.
- [5] Wikipedia: „Atmega328“, s interneta, <https://en.wikipedia.org/wiki/ATmega328>, kolovoz 2024.
- [6] Arduino.cc: „Arduino Uno Rev3 SMD“, s interneta, <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>, kolovoz 2024.
- [7] Wikipedia: „Servomotor“, s interneta, <https://en.wikipedia.org/wiki/Servomotor>, kolovoz 2024.
- [8] Wikipedia: „Pulse-width modulation“, s interneta, https://en.wikipedia.org/wiki/Pulse-width_modulation, kolovoz 2024.
- [9] Wikipedia: „Liquid-crystal display“, s interneta, https://en.wikipedia.org/wiki/Liquid-crystal_display, kolovoz 2024.
- [10] Wikipedia: „I²C“, s interneta, <https://en.wikipedia.org/wiki/I%C2%B2C>, kolovoz 2024.
- [11] Arduino.cc: „Inter-Integrated Circuit Protocol“, s interneta, <https://docs.arduino.cc/learn/communication/wire/#what-is-i2c>, kolovoz 2024.
- [12] Wikipedia: „Membrane keyboard“, s interneta, https://en.wikipedia.org/wiki/Membrane_keyboard, kolovoz 2024.
- [13] Circuit basic: „How to set up a keypad on an arduino“, s interneta, <https://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/>, kolovoz 2024.

[14] Wikipedia: „Sensor“, s interneta, <https://en.wikipedia.org/wiki/Sensor>, kolovoz 2024.

[15] Wikipedia: „Proximity sensor“, s interneta, https://en.wikipedia.org/wiki/Proximity_sensor, kolovoz 2024.

[16] Wikipedia: „Prototipska ploča“, s interneta, https://sh.wikipedia.org/wiki/Prototipska_plo%C4%8Da, kolovoz 2024.

[17] Wikipedia: „Računalno programiranje“, s interneta, https://hr.wikipedia.org/wiki/Ra%C4%8Dunalno_programiranje, kolovoz 2024.

[18] Wikipedia: „Programska knjižnica“, s interneta, https://hr.wikipedia.org/wiki/Programska_knji%C5%BEnica, kolovoz 2024.

[19] Arduino.cc: „Language reference“, s interneta, <https://www.arduino.cc/reference/en/>, kolovoz 2024.

[20] Enciklopedija: „ploče, drvene“, s interneta, <https://www.enciklopedija.hr/clanak/ploce-drvene>, kolovoz 2024.

SAŽETAK I KLJUČNE RIJEČI

U ovom radu izrađen je prototip samoposlužnog automata koji omogućuje automatiziranu i jednostavnu kupovinu proizvoda koji se nalaze unutar njega. Električne komponente koje su potrebne su Arduino Uno, servomotori, LCD zaslon, tipkovnica i senzor blizine čiji rad je objašnjen zasebno i kada čine jednu cjelinu. Programski kod koji je objašnjen i koji upravlja radom navedenih električnih komponenata napisan je u programu Arduino IDE. Kućište je napravljeno od drva i u njemu su smještene sve komponente i proizvodi za prodaju. Uređaj radi na način da korisnici ubacuju kovanicu, odabiru proizvod koji žele i koji uređaj tada ubacuje u prostor gdje ga korisnici mogu jednostavno preuzeti.

Ključne riječi: automatizacija, Arduino, električne komponente, Arduino IDE, drveno kućište

SUMMARY AND KEY WORDS

In this paper, a prototype of a vending machine that enables automated and simple shopping of the products inside it is created. The electrical components that are needed are the Arduino Uno, servo motors, LCD screen, keyboard and proximity sensor, the operation of which is explained separately and when they form a single unit. The program code that is explained and controls the operation of the aforementioned electrical components is written in the Arduino IDE program. The casing is made out of wood and all the components and products for sale are placed within it. The device works in such a way that the users insert a coin, select the product they want, and the device then inserts it into a space where the users can easily pick it up.

Keywords: automation, Arduino, electrical components, Arduino IDE, wooden casing

POPIS SLIKA

Slika 3.1. Arduino Uno R3 SMD	6
Slika 3.2. Arduino Shield povezan na Arduino.....	7
Slika 3.3. Prikaz pinova na pločici Arduino Uno R3 SMD	10
Slika 3.4. Servomotor Tower Pro MG995 360°	12
Slika 3.5. Prikaz modulacije širine impulsa	13
Slika 3.6. Shema spoja servomotora sa Arduino.....	14
Slika 3.7. LCD 16x2.....	16
Slika 3.8. Arduino Uno povezan sa LCD-om	17
Slika 3.9. Arduino Uno povezan sa LCD-om preko IC2 modula	19
Slika 3.10. 3x4 i 4x4 membranska tipkovnica	20
Slika 3.11. Način povezivanja redova i stupaca.....	21
Slika 3.12. FC-51 infracrveni senzor spojen na Arduino Uno	23
Slika 3.13. Eksperimentalna pločica	25
Slika 4.1. Arduino IDE.....	27
Slika 5.1. Korištena ubodna pila	33
Slika 6.1. Iverica2 s utorima za motore.....	40
Slika 6.2. Dio izrade kućišta.....	41
Slika 6.3. Žičane spirale	42
Slika 6.4. Prototip samoposlužnog automata temeljen na Arduino	43
Slika 6.5. Prikaz otvorenih strana samoposlužnog automata	44

POPIS TABLICA

Tablica 3.1. Osnovne karakteristike čipa	8
Tablica 3.2. Pinovi snage	9
Tablica 3.3. Analogni pinovi.....	9
Tablica 3.4. Digitalni pinovi	9
Tablica 3.5. Ostali pinovi	10
Tablica 3.6. Usporedba pozicijskih i kontinuiranih servo motora	14
Tablica 3.7. Oznaka i opis pinova.....	16
Tablica 3.8. Oznake pinova na I2C modulu.....	18
Tablica 6.1. Dimenzije izrezanih ploča.....	40