

# Izrada svjetlosne igre na Arduino platformi

---

**Mrakužić, Antonio**

**Master's thesis / Diplomski rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Split, Faculty of Science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:166:809690>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-11-27**

*Repository / Repozitorij:*

[Repository of Faculty of Science](#)



SVEUČILIŠTE U SPLITU  
PRIRODOSLOVNO MATEMATIČKI FAKULTET

DIPLOMSKI RAD

**IZRADA SVJETLOSNE IGRE NA ARDUINO  
PLATFORMI**

Antonio Mrakužić

Split, rujan 2024.

# Temeljna dokumentacijska kartica

Diplomski rad

Sveučilište u Splitu

Prirodoslovno-matematički fakultet

Odjel za politehniku

Ruđera Boškovića 33, 21000 Split, Hrvatska

## IZRADA SVJETLOSNE IGRE NA ARDUINO PLATFORMI

Antonio Mrakužić

### SAŽETAK

Ovaj rad prikazuje razvoj funkcionalnog sustava igre križić-kružić koristeći Arduino platformu, senzore i LED tehnologiju. Rad započinje pregledom Arduino platforme, s fokusom na model ploče Mega 2560, njegove komponente te postupak programiranja Arduino pločica. Nastavlja se detaljnom analizom senzora, posebno senzora svjetlosti poput fotootpornika, fotodioda i fototranzistora. Također se razmatra LED tehnologija, uključujući princip rada, karakteristike, vrste LED dioda te njihove prednosti u odnosu na klasične izvore svjetlosti. Igra križić-kružić analizira se kroz njena pravila i strategije koje igrači mogu koristiti. Praktični dio rada opisuje zahtjeve sustava, korištene dijelove, proces izrade te implementaciju igre koristeći Arduino, uz vizualni prikaz sustava i izvorni kod. Rad završava procjenom mogućnosti za poboljšanje i proširenje funkcionalnosti sustava.

**Ključne riječi:** Arduino, senzor, LED, križić-kružić

**Rad sadrži:** 67 stranica, 36 grafičkih prikaza, 3 tablice i 36 literaturnih navoda. Izvornik je na hrvatskom jeziku

**Mentor:** **dr. sc. Barbara Džaja**, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

**Neposredni voditelj:** **Hrvoje Turić**, *viši predavač Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

**Ocjenjivači: Dr. sc. Barbara Džaja**, *docent Prirodoslovno-matematičkog fakulteta,  
Sveučilišta u Splitu*

**Hrvoje Turić**, *viši predavač Prirodoslovno-matematičkog fakulteta, Sveučilišta  
u Splitu*

**Dr. sc. Endri Garafulić**, *docent Prirodoslovno-matematičkog fakulteta,  
Sveučilišta u Splitu*

Rad prihvaćen: rujan, 2024. godine

## Basic documentation card

Master thesis

University of Split  
Faculty of Science  
Department of Polytechnics  
Ruđera Boškovića 33, 21000 Split, Croatia

### CREATING A LIGHT GAME ON THE ARDUINO PLATFORM

Antonio Mrakužić

#### ABSTRACT

This paper presents the development of a functional Tic-Tac-Toe game system using the Arduino platform, sensors, and LED technology. The paper begins with an overview of the Arduino platform, focusing on the Mega 2560 model board, its components, and the process of programming Arduino boards. It continues with a detailed exploration of sensors, particularly light sensors such as photoresistors, photodiodes, and phototransistors. LED technology is also examined, covering the operating principles, characteristics, types of LEDs, and their advantages over traditional light sources. The Tic-Tac-Toe game is analyzed through its rules and player strategies. The practical section of the paper outlines the system requirements, components, design process, and implementation of the game using Arduino, with a visual representation of the system and source code. The paper concludes with an assessment of potential improvements and expansions of the system's functionality.

**Keywords:** Arduino, sensors, LED technology, Tic-Tac-Toe

**Thesis consists of:** 67 pages, 36 figures, 3 tables and 36 references. Original language: Croatian

**Mentor:** **Barbara Džaja, Ph.D.** *Assistant Professor of Faculty of Science, University of Split*

**Immediate supervisor:** **Hrvoje Turić,** *Senior Lecturer of Faculty of Science, University of Split*

**Reviewers:** **Barbara Džaja, Ph.D.** *Assistant Professor of Faculty of Science, University of Split*

**Hrvoje Turić**, *Senior Lecturer of Faculty of Science, University of Split*

**Endri Garafulić, Ph.D.** *Assistant Professor of Faculty of Science,  
University of Split*

Thesis accepted: September, 2024

# IZJAVA

kojom izjavljujem s punom materijalnom i moralnom odgovornošću da sam diplomski rad s naslovom IZRADA SVJETLOSNE IGRE NA ARDUINO PLATFORMI izradio samostalno pod voditeljstvom Hrvoja Turića, v.pred. U radu sam primijenio metodologiju znanstveno istraživačkog rada i koristio literaturu koja je navedena na kraju završnog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući naveo u završnom radu na uobičajen, standardan način citirao sam i povezo s fusnotama s korištenim bibliografskim jedinicama. Rad je pisan u duhu hrvatskog jezika.

Student:

Antonio Mrakužić

# Sadržaj

1. Uvod.....	1
2. Arduino .....	2
2.1 Arduino Mega 2560 .....	4
2.1.1 Osnovni dijelovi.....	5
2.2 Programiranje Arduino pločica.....	10
3. Senzori .....	15
3.1 Osnovni pojmovi i principi rada .....	15
3.2 Podjela.....	17
3.3 Senzori svjetlosti.....	19
3.3.1 Fotootpornik.....	19
3.3.2 Fotodioda .....	20
3.3.3 Fototranzistor .....	21
4. LED tehnologija.....	23
4.1 Princip rada LED diode .....	24
4.2 Osnovne karakteristike.....	25
4.3 Vrste LED dioda .....	27
4.4 Prednosti korištenja.....	30
5. Križić-kružić .....	31
5.1 Pravila .....	32
5.2 Strategije .....	32
5.2.1 X1 u kutu.....	33
5.2.2 X1 na stranama.....	35
5.2.3 X1 u sredini .....	38
6. Praktični dio .....	40



6.1 Zahtjevi sustava .....	40
6.2 Opis korištenih dijelova .....	41
6.3 Proces izrade .....	43
6.4 Implementacija.....	46
6.4.1 Dijagram .....	46
6.4.2 Izvorni kod .....	47
6.4.3 Vizualni prikaz sustava .....	51
6.5 Mogućnosti poboljšanja i proširenja funkcionalnosti .....	53
7. Zaključak.....	54
8. Literatura.....	55

# 1. Uvod

Kroz ovaj rad pokazat će se kako se pomoću relativno jednostavnih tehnologija i alata može izraditi jedan funkcionalan sustav. Temeljno razumijevanje elektroničkih komponenti, programiranja i integracije različitih tehnologija ključno je za uspješno kreiranje nekog sustava. U ovom radu realizirana je svjetlosno kontrolirana igra križić-kružić. Iako se radi o jednostavnoj igri, njezina svjetlosno kontrolirana verzija donosi nove izazove vezane za implementaciju logike igre, detekciju korisničkih poteza i vizualizaciju rezultata. Pri realizaciji složenih sustava, kao što je interaktivna igra križić-kružić, moguće je koristiti različite pristupe i tehnologije za postizanje istih ciljeva. Različiti elementi i komponente mogu ispuniti slične funkcionalnosti, a izbor specifičnih tehnologija može utjecati na način implementacije, troškove, i kompleksnost projekta. Kroz iduća poglavlja bit će detaljno opisane korištene tehnologije i komponente realizaciji sustava, kao i detaljni koraci izrade.

U prvom poglavlju bit će obrađena Arduino platforma, razlozi korištenja te neke prednosti u odnosu na druge slične platforme. Također će za potrebe ovog rada biti detaljno opisani dijelovi razvojne pločice odabrane za praktičan rad. Nakon toga, detaljno će biti prikazan postupak programiranja Arduino pločica. Drugo poglavlje obrađuje senzore, osnovne pojmove te različite načine njihove podjele. Za potrebe praktičnog rada detaljnije će biti opisati senzori svjetlosti i njihovi principi rada. Nakon toga, u trećem poglavlju dan je pregled LED tehnologije. Rad govori o osnovnim karakteristikama elemenata koji koriste ovu tehnologiju, njihovom principu rada te prednosti korištenja u odnosu na klasične izvore svjetlosti. U četvrtom poglavlju analizira se igra križić-kružić. Upravo je ona odabrana za realizaciju praktičnog dijela rada. Njezina pravila detaljno su opisana kao i strategije koje igrači mogu primijeniti prilikom igranja. Za kraj, u petom poglavlju zaokružuje se cjelokupna priča s praktičnom realizacijom sustava za igranje igre križić-kružić. Konkretiziraju se zahtjevi sustava i opisuju korišteni dijelovi detaljno analizirani u prethodnim poglavljima. Na kraju se prikazuje proces izrade te i objašnjava programsko rješenje potrebno za ovaj sustav.

## 2. Arduino

Rad s mikroprocesorima i ugrađenim sustavima prije pojave Arduina bio je poprilično zahtjevan. Zbog svoje složenosti korištenja time su se bavili ljudi koji su posjedovali specijalizirano tehničko znanje. Često su se trebali raditi vlastite razvojne pločice za izradu projekata za što većina početnika u elektronici nije imala potrebne vještine. Također, najčešće se programiralo u Asembleru, što je programski jezik niže razine koji je znatno složeniji za učenje od današnjih jezika. Uz to, postojalo je malo alata koji bi omogućili početnicima učenje. Alati su bili nestandardizirani, a platforme neprilagođene korisnicima. Svjesni upravo tih problema studenti jednog talijanskog sveučilišta kreiraju Arduino platformu koja će koristiti za izradu elektroničkih prototipova. Već godinu poslije izdaju svoje prve razvojne pločice: Arduino Uno, Arduino Duemilanove i Arduino Nano. Nedugo zatim napravljen je i Arduino IDE, razvojno okruženje koje omogućuje korisnicima izradu programa koji će se moći pohraniti i izvršavati na Arduinu pločici [1].

Što je zapravo Arduino?

Arduino je elektronička open-source platforma, odnosno platforma otvorenog kôda kojoj je svrha izrada elektroničkih prototipova i konkretnih projekata. Obzirom na to da je *open-source* tipa, svatko ima pristup izvornom kodu i može ga izmijeniti prema vlastitim potrebama. Uz to, fizički dio platforme se također može izmijeniti i nadograditi što omogućava veliku fleksibilnost. Platforma se sastoji od 2 dijela i to [2, 3]:

- Programabilna pločica – naziva se još mikrokontroler zbog glavnog elementa pločice
- Programska podrška – integrirano razvojno okruženje (*engl. Integrated Development Environment*), najčešće Arduino IDE

Zbog svoje jednostavnosti Arduino se danas koristi u raznim područjima. Neki od njih su [4]:

- Edukacija
- Internet stvari (*engl. Internet of Things*)
- Robotika

- Pametne kuće
- Medicina
- Automatizirana vozila

Postoje razne druge platforme koje obavljaju istu ili sličnu zadaću kao i Arduino, međutim Arduino ima nekoliko prednosti zbog kojih je stekao veliku popularnost među nastavnicima, učenicima i početnicima u elektronici. Prednosti su sljedeće [5]:

- Cijena
- Dostupnost na više platformi – dostupan je za Windows, Mac i Linux operativne sustave
- Jednostavnost korištenja – pogodno za početnike, nije potrebno puno predznanja za kreirati početne projekte
- Open-source – Sva proširenja i biblioteke su dostupne svima te ih korisnik može prilagoditi svojim potrebama
- Laka nadogradnja – osnovnim pločicama se lako mogu proširiti mogućnosti pomoću štitova (*engl. Shield*)

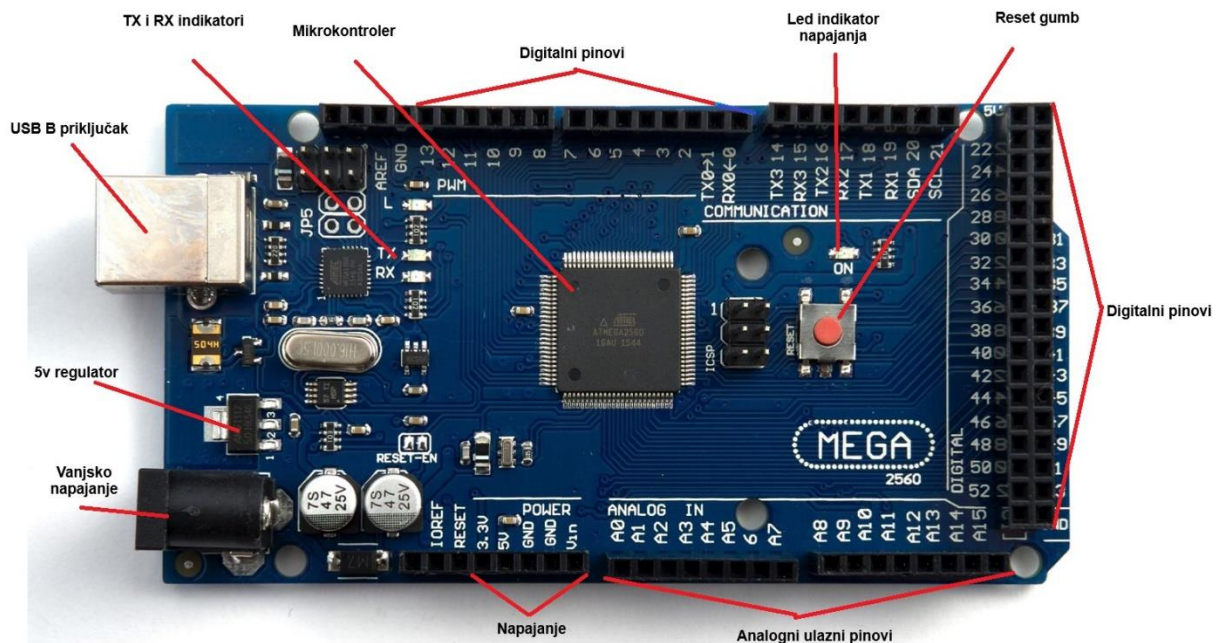
Ovisno o specifičnostima projekta, korisnik ima na izbor veliki broj Arduino pločica. Neke od razvojnih pločica su:

- Arduino Uno
- Arduino Nano
- Arduino Micro
- Arduino Duo
- Arduino Mega

U nastavku će detaljno biti opisana Arduino Mega 2560 pločica, koja je odabrana za rad zbog specifičnih zahtjeva praktičnog projekta.

## 2.1 Arduino Mega 2560

Arduino Mega 2560 jedna je od najmoćnijih i najpopularnijih pločica u Arduino obitelji, namijenjena projektima koji zahtijevaju veći broj ulaza i izlaza, veću procesorsku snagu te dodatne mogućnosti proširenja. S obzirom na njezine mogućnosti i fleksibilnost, Arduino Mega 2560 pruža čvrstu osnovu za razvoj naprednih elektroničkih projekata koji nadilaze osnovne funkcionalnosti manjih Arduino pločica.



**Slika 1.** Arduino Mega 2560 razvojna pločica [6]

Slika 1 prikazuje Arduino Mega 2560 ploču s označenim osnovnim dijelovima koji će detaljnije biti opisani u sljedećem poglavlju.

Tehničke specifikacije za ovu ploču se značajno razlikuju u odnosu na osnovnu Arduino Uno pločicu. Neke od najbitnijih razlika su količina memorije, te broj pinova dostupnih korisnicima.

Osnovne tehničke specifikacije prikazane su u Tablici 1 [7]:

**Tablica 1.** Tehničke specifikacije Arduino Mega 2560

<b>Specifikacija</b>	<b>Opis</b>
<b>Mikrokontroler</b>	ATmega2560
<b>Arhitektura</b>	AVR
<b>Radni napon</b>	5V
<b>Ulazni napon</b>	6V – 20V
<b>Takt</b>	16 MHz
<b>Flash memorija</b>	256 KB
<b>SRAM</b>	8 KB
<b>EEPROM</b>	4 KB
<b>Digitalni pinovi</b>	54
<b>Analogni pinovi</b>	16
<b>Maksimalna struja po pinu</b>	40mA

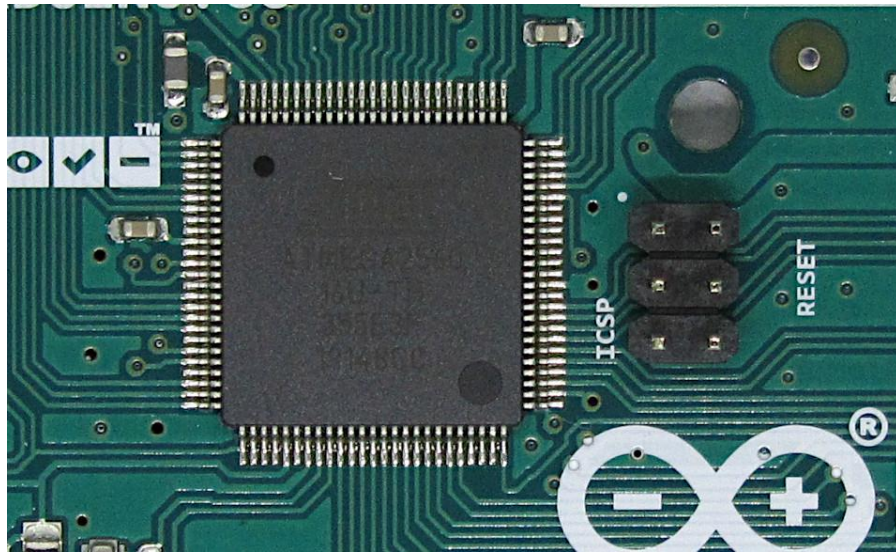
### **2.1.1 Osnovni dijelovi**

U ovom poglavlju bit će opisani glavni dijelovi Arduino Mega 2560 ploče i detaljno analizirane njihove funkcije.

#### **2.1.1.1 Mikrokontroleri i memorija**

Osnovni dio Arduino pločice je mikrokontroler. Mikrokontroler, pojednostavljeno rečeno, je malo računalo na jednom čipu. Dizajniran je kako bi upravljao konkretnim zadacima unutar ugrađenog sustava bez potrebe za kompleksnijim operacijskim sustavima. Za razliku od mikroprocesora opće namjene, mikrokontroleri spajaju procesorsku jedinicu, memoriju i ulazno izlazne jedinice, koje uključuju analogno-digitalne pretvornike i brojače, sve u jednu učinkovitu i povoljnu samostalnu jedinicu. Spajajući nekoliko komponenti u jedan sustav, mikrokontroleri su prikladni za aplikacije koje zahtijevaju obradu signala u realnom vremenu kao što su upravljanje motorima ili nekim drugim aktuatorima pomoću raznih senzora [8].

Arduino Mega posjeduje 2 mikrokontrolera. Prvi, glavni mikrokontroler, pod nazivom ATmega2560, pokreće korisničke programe koji se putem Arduino IDE prebacuju na Arduino pločicu. Prema njemu je Arduino pločica dobila ime. Slika 2 prikazuje fizički izgled ATmega2560 mikrokontrolera.



**Slika 2.** ATmega2560 mikrokontroler

Sva memorija se nalazi na ovom mikrokontroleru. Vrsta memorije koju posjeduje je:

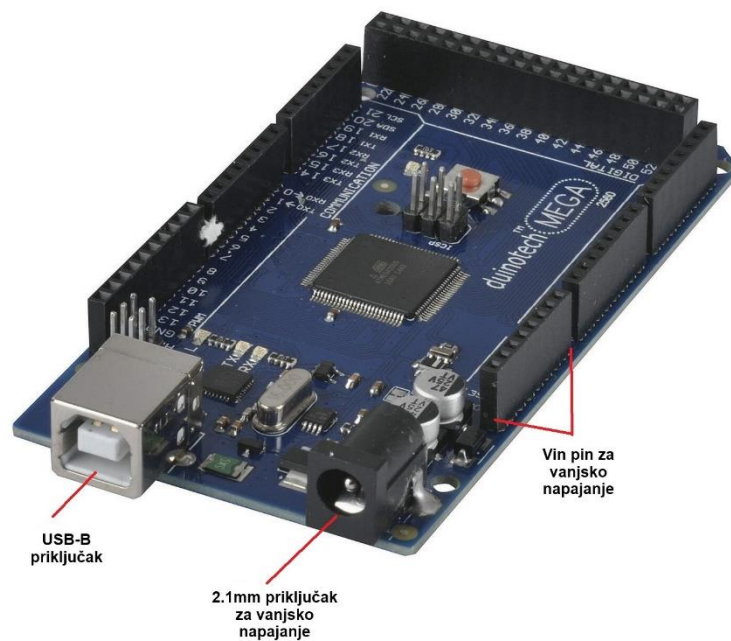
- Flash memorija – za pohranu korisničkog koda
- SRAM (engl. Static Random Access Memory) – za izvršavanje programa
- EEPROM (engl. Electrically Erasable Programmable Read-Only Memory) - koristi se za dugotrajno čuvanje podataka čak i kada mikrokontroler nije pod napajanjem

Drugi mikrokontroler, pod nazivom ATmega16U2 zadužen je za komunikaciju između računala i glavnog mikrokontrolera putem USB priključka.

### 2.1.1.2 Napajanje

Nadalje, kako bi Arduino mogao raditi, potrebno mu je napajanje. Ovisno o zahtjevima, Arduino se može napajati putem USB priključka, vanjskog istosmjernog izvora napajanja ili putem baterije [9].

Slika 3 prikazuje sve mogućnosti napajanja Arduino Mega 2560 ploče.



**Slika 3.** Priključci za napajanje na Arduino MEGA 2560 [9]

Napajanje za Arduino MEGA 2560 može se osigurati putem USB veze s računalom, koja osigurava 5V. Ovaj način napajanja je prikladan za prototipove i eksperimentalnu upotrebu.

USB 2.0 priključak na računalo osigurava 5V s tolerancijom od  $\pm 5\%$  i može isporučiti do 500mA struje. Arduino MEGA 2560 ploče imaju ugrađen resetirajući osigurač od 500mA koji štiti 5V napajanje s USB priključka. Ovaj osigurač prekida strujni krug u slučaju preopterećenja ili kratkog spoja, ali se automatski resetira nakon što se problem ukloni. Važno je napomenuti da se mali dio USB napona od 5V gubi na osiguraču, a taj pad napona raste kako se povećava struja kroz osigurač,



što rezultira smanjenjem napona na 5V pinu Arduina. Ovo je posebice bitno uzeti u obzir u slučaju da je više elemenata spojeno na Arduino jer postoji mogućnost pojave poteškoća prilikom komunikacije između računala i pločice. Osim što omogućuje napajanje Arduino MEGA 2560 ploče, USB veza također omogućuje programiranje Arduina putem Arduino IDE-a. Za povezivanje Arduino MEGA 2560 s računalom koristi se običan USB kabel za printere. Ova vrsta kabela ima USB priključak tipa A na strani računala i tipa B na strani Arduina. Takvi se kablovi nazivaju A-B USB kablovi.

Vanjsko napajanje može se osigurati putem priključka koji se nalazi pored USB priključka na Arduino Mega 2560 ploči, omogućujući ploči da radi samostalno, bez potrebe za USB vezom s računalom. Preporučuje se vanjsko istosmjerno (engl. DC, Direct Current) napajanje s naponom od 7V do 12V, koje može isporučiti barem 500mA struje. Središnji pin konektora mora biti povezan s pozitivnim priključkom vanjskog napajanja. Stvarna potreba za strujom ovisi o projektu u kojem se Arduino koristi. Arduino sam po sebi troši malo struje, dok se ostatak povlači iz strujnog kruga ili štita koji je na njega spojen. Ako je USB kabel priključen na Arduino MEGA 2560 za programiranje i istovremeno je spojeno vanjsko napajanje, Arduino će automatski koristiti vanjsko napajanje. Elektronika za napajanje prepoznaje kada je vanjsko napajanje spojeno i automatski se prebacuje na njega. Arduino se također može napajati putem vanjskog izvora koristeći Vin i Gnd pinove na pločici.

Posljednji način napajanja Arduino razvojne pločice je putem baterije. Baterija se može spojiti na vanjski 2,1 mm priključak za napajanje, kao i prethodno spomenuto vanjsko napajanje, ili putem Vin pina na Arduino MEGA 2560 ploči. Prilikom povezivanja baterije, treba pripaziti na polaritet veze. Potrebno je spojiti središnji pin priključka za napajanje na pozitivni pol baterije, a negativni pol baterije na Gnd. Ako se baterija spaja na Vin pin na zaglavlju Arduino MEGA 2560, potrebno je povezati pozitivni pol baterije s Vin pinom, a negativni s Gnd pinom.

Uobičajeni naponi baterija koji su sigurni za upotrebu s Arduino MEGA 2560 putem priključka za napajanje ili Vin pina su 9V i 12V. Ovaj način napajanja Arduina se uglavnom koristi prilikom jednostavnijih projekata jer baterije imaju ograničen kapacitet, što znači da ne mogu dugotrajno napajati pločicu [9].

### 2.1.1.2 Pinovi

Nadalje, potrebno je objasniti pinove koje korisnik može koristiti prilikom izrade svojeg projekta. Svrha pinova je raznolika, a za potrebne ovog rada bit će objašnjeni oni koji se najčešće koriste. Neki od najvažnijih pinova su [9]:

- Pinovi za napajanje
- Analogni ulazni pinovi
- Digitalni pinovi
- Reset pin

Pinovi za napajanje su sljedeći:

- Vin – služi za napajanje ploče s vanjskim izvorom napona (7-12V)
- 5V – ovaj pin dobiva napon od USB-a, regulatora koji se nalazi na ploči ili vanjskog izvora. može se koristiti za štitove ili ostale strujne krugove
- 3.3V – koristi se za strujne krugove za koje je potrebno niže napajanje od 5V, a dobiva se pomoću regulatora na pločici
- Gnd – pinovi za uzemljenje; Arduino Mega 2560 ima nekoliko ovih pinova za povezivanje s različitim komponentama

Arduino Mega 2560 sadrži 16 analognih pinova označenih od A0 do A15. Oni se koriste za čitanje analognih signala. Spojeni su na 10-bitni analogno-digitalni pretvornik koji koristi 5V kao referentni napon pomoću kojeg se određuje iznos napona na analognom pinu. Taj pretvornik ima mogućnost dobivanja  $2^{10}$ , odnosno 1024 stanja, ovisno o razini napona. Na ovim pinovima se očitavaju vrijednosti dobivene od senzora koje će kasnije poslužiti za kontrolu toka programskog koda.

Osim analognih, Arduino Mega ima i 54 digitalna pina, označenih od 0 do 52, koji se mogu koristiti kao ulazni ili izlazni pinovi. Digitalne pinove postavi se kao ulazne pinove ako se žele očitavati jednostavna stanja poput uključeno/isključeno, prisutno/odsutno. Ovi ulazi očitavaju podatke u obliku logičkih signala, odnosno visok napon predstavljen je sa 1 a niski napon sa 0. Digitalni

pinovi mogu služiti kao izlazni ako je potrebno kontrolirati neke vanjske komponente poput LED, releja, motora ili LCD ekrana.

Reset pin služi za ponovno pokretanje glavnog mikrokontrolera i najčešće se koristi kad glavni reset gumb na Arduino nije dostupan. Pogodan je za korištenje ako postoji potreba za ponovnim pokretanjem programa učitano na pločicu.

### 2.1.1.3 LED indikatori

Osim pinova potrebno je spomenuti i LED indikatore koji pružaju dodatne informacije korisniku. LED indikatori koji se nalaze na Arduino Mega 2560 pločici te njihova funkcija navedeni su u Tablici 2.

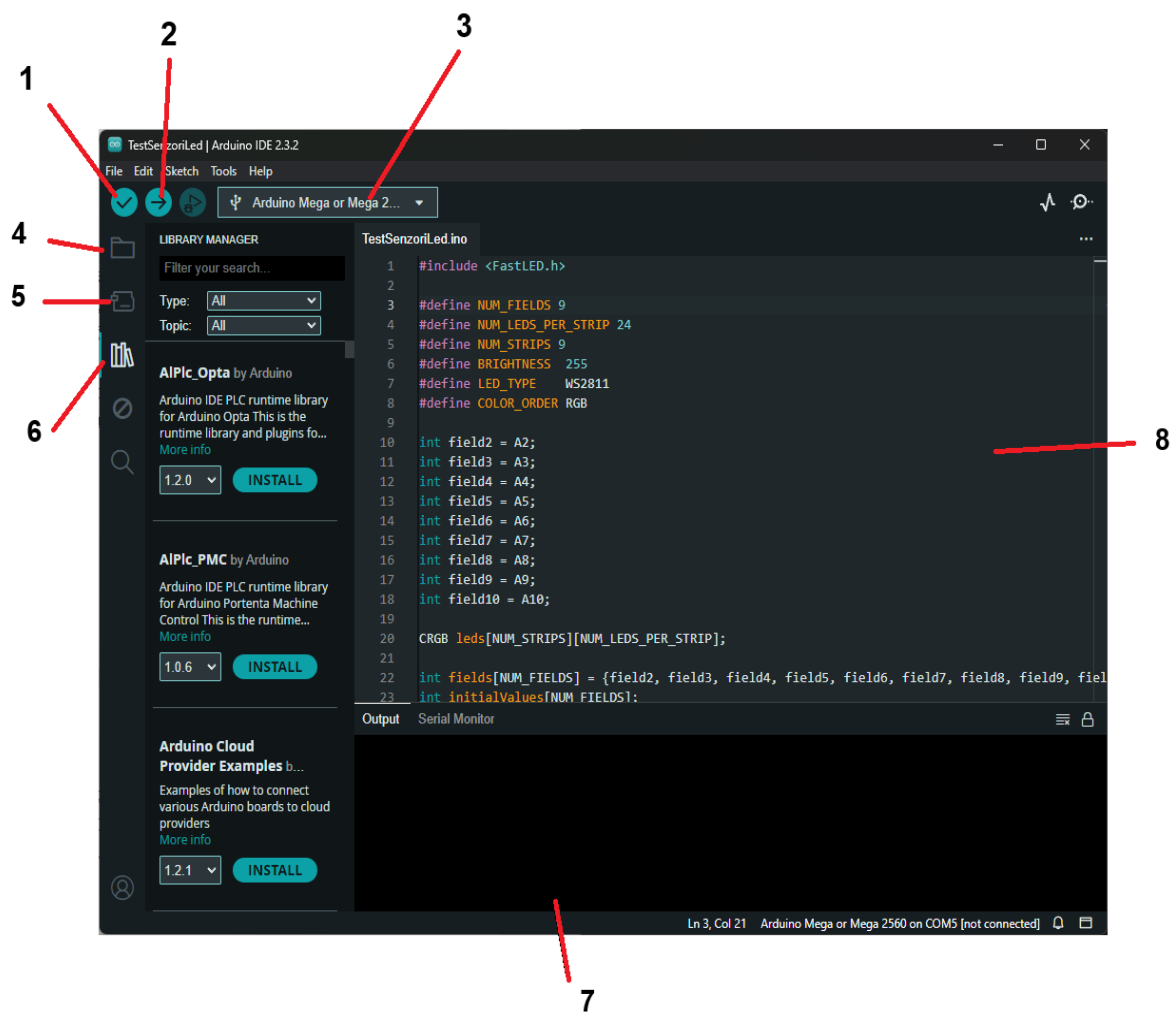
**Tablica 2.** LED indikatori na Arduino Mega 2560 ploči [9]

Ime LED indikatora	Opis
ON	Svijetli kad je Arduino Mega 2560 pločica upaljena.
TX	Treperi kad Arduino odašilje podatke.
RX	Treperi kad Arduino prima podatke.
L	Programabilna LED. Korisnik može upravljati ovom LED-icom putem koda

## 2.2 Programiranje Arduino pločica

Arduino je popularna platforma koja omogućava jednostavan pristup elektronici i programiranju, čak i za početnike. Ova platforma se sastoji od hardverskog dijela, koji uključuje različite vrste Arduino pločica, i softverskog dijela koji je baziran na Arduino IDE-u. Ovo poglavlje posvećeno je osnovama programiranja Arduino pločica: kako započeti s radom, kako povezati pločicu s

računalom, osnovna struktura Arduino programa te tipične funkcije koje se koriste u razvoju aplikacija.



Slika 4. Arduino razvojno okruženje

Prvi korak u programiranju Arduino pločica je instalacija Arduino IDE-a. Arduino IDE je integrirano razvojno okruženje koje omogućava pisanje, kompajliranje i učitavanje koda na Arduino pločicu. IDE je dostupan za različite operativne sustave, uključujući Windows, macOS i Linux. Arduino IDE se može preuzeti sa službene Arduino stranice te zatim instalirati prema uputama za određeni operativni sustav. Za pojedince koji preferiraju rad u pregledniku, postoji i

opcija korištenja Arduino Web IDE-a. Ova verzija ne zahtijeva instalaciju i omogućava spremanje projekata u oblaku, čime postaju dostupni s bilo kojeg uređaja [10].

Glavni dijelovi programa označeni su na Slici 4 brojevima od 1 do 8, a to su redom:

- Gumb za kompajliranje skice, odnosno trenutnog programa
- Gumb za prebacivanje programa sa računala na mikrokontroler Arduino ploče
- Padajući izbornik za odabir Arduino ploče i porta koji će se koristiti
- Prikaz svih skica
- Upravitelj ploča – služi za instalaciju programske podrške za ispravno korištenje Arduino ploča
- Upravitelj bibliotekama – služi za instalaciju biblioteka koje se kasnije koriste prilikom izrade programa
- Prostor za prikaz stanja programa; koristi se za ispis pogrešaka, uspješnost kompajliranja ili ispis stanja učitano programa putem serial monitora
- Dio programa namijenjen za pisanje koda

Nakon instalacije Arduino IDE-a, sljedeći korak je povezivanje Arduino pločice s računalom. Većina Arduino pločica, dolazi s ugrađenim USB portom koji se koristi za povezivanje s računalom. Za povezivanje, jednostavno se priključi USB kabel s jedne strane u računalo, a s druge strane u odgovarajući USB port na pločici. Nakon povezivanja, Arduino IDE bi trebao automatski prepoznati pločicu. Ako se to ne dogodi, može se ručno odabrati odgovarajuća pločica i port unutar sučelja [10].

Arduino programi, poznati kao skice, pišu se u programskom jeziku koji je baziran na C/C++. Svaki Arduino program mora sadržavati dvije osnovne funkcije [11]:

- `setup()`: Ova funkcija se izvršava samo jednom, odmah nakon što se pločica uključi ili resetira. U njoj se obično postavljaju inicijalne konfiguracije poput definiranja načina rada pinova (ulaz/izlaz), pokretanja serijske komunikacije i inicijalizacije biblioteka.

- `loop()`: Nakon funkcije `setup()`, `loop()` se izvršava kontinuirano sve dok je pločica uključena. Ovdje se nalazi glavni dio programa, gdje se obavljaju ponavljajuće operacije kao što su čitanje podataka sa senzora, kontrola izlaznih uređaja i slanje podataka putem serijske komunikacije.

Postoji nekoliko osnovnih funkcija koje se često koriste u Arduino programiranju [12]:

- `pinMode()`: Postavlja način rada određenog pina kao ulaznog (*engl. input*) ili izlaznog (*engl. output*).
- `digitalWrite()`: Postavlja izlazni pin na visoku (*engl. high*) ili nisku (*engl. low*) vrijednost.
- `digitalRead()`: Čita trenutnu vrijednost ulaznog pina.
- `analogRead()`: Čita analogni ulaz na određenom pinu (vrijednosti od 0 do 1023).
- `delay()`: Pauzira izvršavanje programa na određeni broj milisekundi.
- `Serial.begin()`: Inicijalizira serijsku komunikaciju na određenoj brzini prijenosa podataka.
- `Serial.print()` i `Serial.println()`: Ispisuje podatke putem serijskog porta, s time da `println()` dodaje novi red nakon ispisa.

Nadalje, potrebno je spomenuti biblioteke. Biblioteke u Arduinu su važni dodaci koji proširuju osnovne mogućnosti Arduino programa, a sastoje se od tisuća službenih i korisnički razvijenih biblioteka. Njihova važnost u programiranju Arduina je ogromna jer značajno pojednostavljaju upotrebu inače složenog koda. Na primjer, umjesto da sami pišete kod za čitanje specifičnih senzora, upravljanje motorima ili povezivanje na internet, jednostavno možete instalirati odgovarajuću biblioteku, uključiti je na početku svog koda i iskoristiti sve dostupne funkcionalnosti. Sve Arduino biblioteke su *open source* i besplatne za korištenje, što ih čini neprocjenjivim alatom za programere na svim razinama.

Tipični proces za programiranje i učitavanje koda na Arduino pločicu sastoji se od nekoliko koraka [9-11]:

- Instalacija pločice: Ovisno o tome koja se pločica trenutno koristi, možda će trebati instalirati odgovarajući paket unutar Arduino IDE-a.
- Pisanje skice.
- Kompajliranje skice: Prije nego što se kod može učitati na pločicu, potrebno ga je kompajlirati. Ovo uključuje provjeru postoje li greške u pisanju koda.
- Učitavanje skice na pločicu: Nakon uspješnog kompajliranja, odabire se odgovarajući serijski port u IDE-u i učitava program na pločicu.
- Praćenje rada pomoću serijskog monitora: Ako je potrebno, može se koristiti serijski monitor u IDE-u za praćenje izlaznih podataka ili dijagnosticiranje problema.

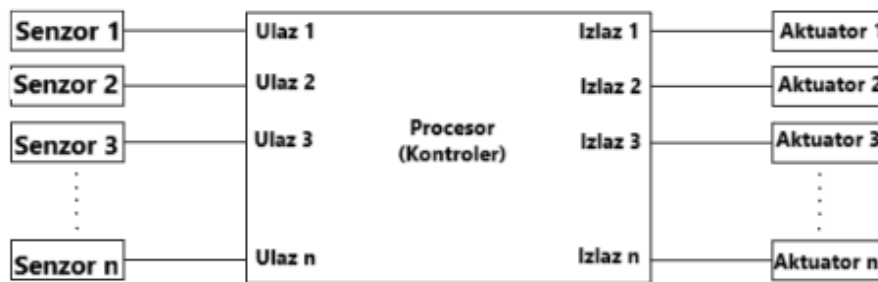
Arduino je izuzetno fleksibilna platforma koja omogućava brzo učenje i razvoj elektronike, čak i bez prethodnog iskustva. Dostupnost osnovnih funkcija i strukture programa omogućuje brzo građenje projekata i proširivanje znanja. Programiranje Arduino pločice ima stvarni smisao tek kada se na nju povežu dodatni elementi, poput senzora i aktuatora. Sama pločica, bez ikakvih povezanih komponenti, ne može ispuniti svoju svrhu. Tipičan način korištenja Arduina uključuje dobivanje informacija iz okoline putem senzora, a zatim, u skladu s definiranom logikom, upravljanje izlaznim elementima poput LED-ica, motora, zvučnika ili drugih uređaja. Na taj način, Arduino omogućava interakciju s fizičkim svijetom, pretvarajući jednostavne ideje u funkcionalne projekte.

## 3. Senzori

Senzor je uređaj koji detektira fizičke promjene u okolini i pretvara te promjene u signal koji se može mjeriti, analizirati i dalje obraditi. Fizičke promjene koje senzor može detektirati uključuju temperaturu, svjetlost, tlak, zvuk, vlažnost, gibanje, magnetsko polje i mnoge druge [13].

### 3.1 Osnovni pojmovi i principi rada

Uloga senzora je pretvaranje neke fizičke veličine u električni signal. Ovaj signal može biti analogni ili digitalni te se koristi kao ulazni podatak za procesore i druge dijelove sustava. Na temelju signala koji dolazi iz senzora, sustavi mogu automatski reagirati ili prilagoditi svoje ponašanje. Senzori su najčešće dio nekih većih sustava koji uz njih koriste i aktuatore, upravljačke jedinice kao i neke pomoćne komponente poput napajanja. Slika 5 prikazuje ulogu senzora u kontekstu nekog sustava.



Slika 5. Prikaz senzora u složenom sustavu [14]

Senzor se može gledati kao ulaz u sustav, dok aktuator kao izlaz. Točnije, senzor kao ulaz daje signal učitani iz okoline, upravljačka jedinica ga obrađuje i sukladno logici aktuator obavlja neku fizičku radnju.

Da bi senzor uspješno obavljao svoju funkciju mora sadržavati sljedeće elemente [15]:

- Sensorni element – dio senzora koji reagira na promjene u okruženju kao što su svjetlost, temperatura i slično. Primjer senzornog elementa je fotootpornik koji mijenja vrijednost otpora ovisno o količini svjetlosti



- Pretvarač – pretvara promjenu koju je senzorni element uočio u električni signal koji se može izmjeriti i dalje obrađivati
- Obrađivač signala – predstavlja nekoliko vrsta elektroničkih komponenti koje filtriraju, pojačavaju, kalibriraju te pretvaraju analogni signal u digitalni i obratno. Svrha im je prilagodba i poboljšanje signala kako bi on bio što prikladniji za daljnje korištenje
- Izlazno sučelje – šalje obrađeni signal na računalo ili neki drugi sustav

Neke od osnovnih karakteristika koje je potrebno uzeti u obzir prilikom korištenja senzora mogu se podijeliti na dinamičke i statičke.

Statičke karakteristike senzora ukazuju na to koliko sensor ispravno predstavlja mjerenu veličinu nakon što se signal stabilizira. Važne statičke karakteristike senzora uključuju osjetljivost, razlučivost, linearnost, raspon, ponovljivost i reproducibilnost. Osjetljivost ukazuje koliko se izlaz senzora mijenja s promjenom mjerene veličine. To je statička karakteristika jer se odnosi na stabilno stanje nakon što promjena mjerene veličine uzrokuje promjenu izlaza. Razlučivost predstavlja najmanju promjenu mjerene veličine koju sensor može detektirati u stabilnim uvjetima. Linearnost govori koliko je izlaz senzora proporcionalan promjenama mjerene veličine u statičkom stanju. Idealno, sensor bi trebao imati linearan odnos između ulaza i izlaza. Raspon je statička karakteristika koja definira minimalne i maksimalne vrijednosti koje sensor može izmjeriti u stabilnim uvjetima. Ponovljivost je sposobnost senzora da daje iste rezultate pri ponovljenim mjerenjima istih uvjeta u stabilnom stanju. Reproducibilnost predstavlja također sposobnost senzora da daje iste rezultate pri ponovljenim mjerenjima, ali u različitim uvjetima [16] [17].

Dinamičke karakteristike opisuju kako sensor reagira na promjene mjerenih veličina u vremenskom okviru, to jest, koliko brzo i precizno sensor može pratiti promjene. Neke od najbitnijih dinamičkih karakteristika su mrtvo vrijeme, mrtva zona senzora i vrijeme odziva. Mrtvo vrijeme predstavlja vremenski razmak od trenutka kad se mjerena vrijednost stvarno promijeni do trenutka kad se ta promjena iskaže na izlazu senzora. Mrtva zona senzora predstavlja najveću promjenu mjerene veličine do koje može doći, a da još ne dođe do promjene izlaznog signala iz senzora. Vrijeme odziva senzora odnosi se na vrijeme koje je potrebno senzoru da odgovori na promjenu ulaza i stabilizira se na novu vrijednost [16] [17].

## 3.2 Podjela

Senzori se mogu podijeliti na različite načine, ovisno o njihovim karakteristikama, načinu rada, primjeni i fizičkim fenomenima koje detektiraju [18].

Podjela prema načinu napajanja:

- Aktivni - zahtijevaju vanjski izvor energije kako bi radili. Ova energija se koristi za aktivaciju senzora i za generiranje signala.
- Pasivni - ne zahtijevaju vanjski izvor energije za rad. Oni koriste prirodne ili postojeće izvore energije iz okoliša.

Podjela prema načinu kontakta:

- Beskontaktni
- Kontaktni

Podjela prema vrsti izlaznog signala

- Analogni - Signal koji izlazi iz analognih senzora je neprekidan i može imati beskonačno mnogo vrijednosti unutar određenog raspona.
- Digitalni - signal u obliku diskretnih vrijednosti, obično kao binarni broj

U Tablici 3 napravljena je detaljna podjela senzora prema primjeni.

**Tablica 3.** Podjela senzora prema primjeni [18]

<b>Vrsta</b>	<b>Opis</b>	<b>Primjer</b>
<b>Senzori položaja</b>	Mjere fizički položaj objekta.	Potencijometar, Linearni enkoder
<b>Senzori tlaka i sile</b>	Mjere tlak, silu ili deformaciju na objektima.	Barometar, tenzometrijska traka

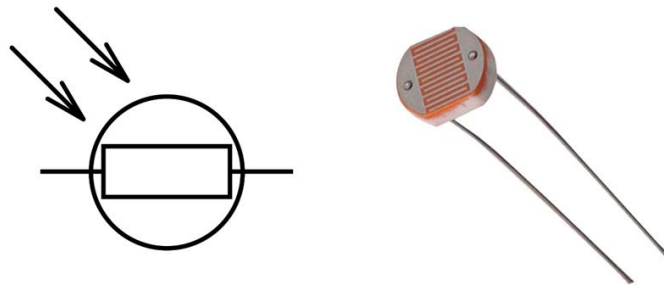
<b>Senzori pokreta</b>	Mjere različite vrste kretanja objekta, uključujući ubrzanje i rotaciju.	PIR senzor, akcelerometar, žiroskop
<b>Senzori slike</b>	Mjere i obrađuju slike. Koriste se u kamerama, skenerima i optičkim sustavima za prepoznavanje i analizu vizualnih informacija.	CCD senzor, CMOS senzor
<b>Optički uređaji</b>	Mjere svjetlost ili svjetlosne promjene. Koriste se u aplikacijama poput osvjetljenja, detekcije svjetlosti i optičkih komunikacija.	Fotodioda, Fotootpornik, Fototranzistor
<b>Magnetni senzori</b>	Mjere promjene u magnetskom polju	Hallov senzor
<b>Kemijski senzori</b>	Mjere kemijske promjene u okruženju. Koriste se za analizu kemijskog sastava, pH vrijednosti i detekciju plinova.	pH senzor, plinski senzor
<b>Akustični senzori</b>	Mjere zvučne valove ili vibracije. Koriste se u komunikaciji, medicinskim uređajima i analizi zvuka.	Mikrofon, ultrazvučni senzor
<b>Termalni senzori</b>	Mjere toplinsku energiju i temperature.	Termistor, infracrveni senzor

### 3.3 Senzori svjetlosti

Obzirom da će se za potrebe praktičnog rada koristiti vrsta senzora koja mjeri razinu svjetlosti i svjetlosnu promjenu potrebno je detaljnije objasniti neke od najčešćih senzornih elemenata koji imaju navedenu funkciju. Neki od najčešćih elektroničkih elemenata koji se koriste u takvim sensorima su fotootpornik, fotodioda i fototranzistor.

#### 3.3.1 Fotootpornik

Fotootpornik je vrsta otpornika koja mijenja iznos svog otpora ovisno o intenzitetu svjetlosti koja ga obasjava. Zbog toga se još naziva LDR (engl. *Light Dependent Resistor*). Ako se nalazi u mraku vrijednost njegovog otpora bit će u megaomima. Međutim, Ako je osvjetljen, ta vrijednost može pasti i do nekoliko stotina oma. Slika 6 prikazuje simbol fotootpornika te njegov fizički izgled [19].

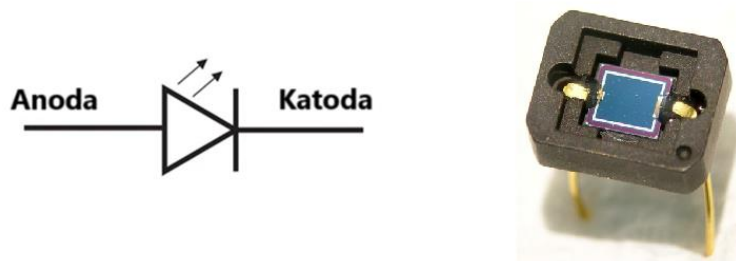


**Slika 6.** Simbol fotootpornika (lijevo) i njegov fizički izgled (desno) [19]

Zbog svojih karakteristika često se koriste za mjerenje intenziteta svjetlosti i prilagodbu svjetlosnih izvora prema uvjetima osvjetljenja. Fotootpornici se izrađuju od posebne vrste poluvodiča, kao što je kadmij sulfid ili olovo sulfid. Kada se ovaj poluvodič stavi u mrak, elektroni unutar njegove strukture ne žele teći kroz otpornik jer su previše vezani za atome kristala. Međutim, kada su osvjetljeni, nadolazeći fotoni sudaraju se s vezanim elektronima, odvajajući ih od atoma, stvarajući tako šupljinu u procesu. Ovi oslobođeni elektroni sada mogu pridonijeti struji koja teče kroz uređaj te se otpor smanjuje [13].

### 3.3.2 Fotodioda

Fotodioda je poluvodički element koji pretvara svjetlost u električnu struju. Sastoji se od tankog n-tipa poluvodiča koji sadrži višak elektrona i debljeg p-tipa poluvodiča koji sadrži višak šupljina. N-strana je katoda, a p-strana anoda. Slika 7 prikazuje simbol fotodiode i fizički izgled jedne od njezinih izvedbi.



**Slika 7.** Simbol fotodiode (lijevo) i fizički izgled (desno) [20]

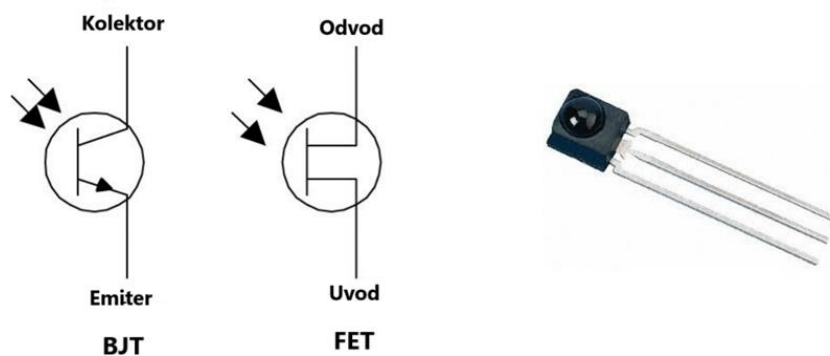
Kada svjetlost padne na fotodiodu, fotoni prolaze kroz n-tip poluvodič i ulazi u p-tip poluvodiča. Tamo se fotoni sudaraju sa vezanim elektronima i izbacuju ih iz atoma, stvarajući rupe. Posljedica gomilanja naboja na suprotnim stranama PN spoja je razlika u potencijalu između anode i katode. Ta pojava se naziva fotoelektrični efekt. Ako spojimo fotodiodu u strujni krug sa trošilom teći će struja od anode prema katodi [13].

Fotodioda može imati dva načina rada [13]:

- Fotonaponski način rada – fotodioda direktno pretvara svjetlost u električnu struju te je ovisnost ulaznog osvjetljenja o izlaznoj struji gotovo linearna.
- Fotovodljivi način rada – fotodioda je polarizirana vanjskim naponskim izvorom, obično u obrnutoj polarizaciji. Kada je fotodioda u mraku, teče vrlo mala struja poznata kao struja manjinskih nositelja naboja. Kada se fotodioda osvjetli, povećava se broj generiranih elektrona i šupljina, što rezultira povećanjem struje kroz fotodiodu. U ovom režimu, fotodioda se ponaša kao otpornik čija otpornost varira s intenzitetom svjetlosti. Veća svjetlost dovodi do veće struje, dok manja svjetlost rezultira manjom strujom.

### 3.3.3 Fototranzistor

Fototranzistor je poluvodički element koji kombinira funkcionalnost fotodiode i tranzistora. Sličan je fotiodiodi po tome što može pretvoriti svjetlosnu energiju u električnu, ali se razlikuje po tome što koristi pojačanje tranzistora za povećanje signala. Slika 8 prikazuje simbole BJT i FET tranzistora te njegov fizički izgled.



**Slika 8.** Simboli BJT i FET fototranzistora (lijevo) i njegov fizički izgled (desno) [21]

Fototranzistor je vrsta tranzistora čija se baza u bipolarnom tranzistoru ili vrata u tranzistoru sa efektom polja (*engl. FET, Field-Effect Transistor*) koriste za detekciju svjetlosti. Kada je izložen svjetlosti, fototranzistor generira električnu struju koja ovisi o intenzitetu svjetlosti. U osnovi, fototranzistor ima dva glavna dijela, a to su fotoosjetljiva regija koja reagira na svjetlost i tranzistorski dio koji pojačava generirani signal. Kada svjetlost padne na fototranzistor, fotoni upadaju u fotoosjetljivi dio uređaja. Ova svjetlost uzrokuje stvaranje parova elektrona i šupljina.

Kod bipolarnog fototranzistora generiranje nositelja naboja može se događati na nekoliko načina, ovisno o konstrukciji fototranzistora. Osnovni način je osvjetljivanje baze fototranzistora gdje svjetlost generira parove elektrona i šupljine. U NPN bipolarnom fototranzistoru, koji se najčešće koristi, baza će biti napravljena od P-tipa poluvodiča. Elektroni u P-tipu poluvodiča, odnosno u ovom slučaju u bazi, su manjinski nositelji, dok su šupljine većinski. Zbog polarizacije PN-spojeva NPN tranzistora generirani elektroni migriraju prema kolektoru, dok će se šupljine kretati prema emiteru. Povećanja koncentracija elektrona u bazi smanjuje barijeru između emitera i baze omogućujući veću emitorsku struju  $I_E$ . Povećanje emitorske struje  $I_E$  znači da će više elektrona

dolaziti do baze, a povećanje bazne struje  $I_B$  prouzročit će značajno povećanje kolektorske struje  $I_C$  zbog tranzistorskog učinka pojačanja. Ako je pak osvijetljen emiter, svjetlost će generirati dodatne slobodne elektrone, povećavajući emitorsku struju  $I_E$ . Povećanje emitorske struje znači da će više elektrona prelaziti u bazu. To povećava baznu struju  $I_B$  koja će omogućiti povećanje kolektorske struje  $I_C$ . Za kraj, ako se osvijetli kolektor, generirani nositelji naboja će odmah se pridodati postojećoj struji kolektora  $I_C$ , u osnovi omogućavajući fototranzistoru bržu reakciju na svjetlost. [22]

Kod FET fototranzistora postoje dva principa, ovisno govori li se o MOSFET-u (*engl. Metal-Oxide-Semiconductor FET*) ili JFET-u (*engl. Junction FET*). U MOSFET fototranzistoru, prisutnost svjetlosti generira dodatne nositelje naboja, povećavajući vodljivost kanala. Ova povećana vodljivost smanjuje otpornost kanala i omogućuje veću struju  $I_D$  između odvoda i izvora. Također, svjetlost smanjuje potreban napon na vratima za održavanje određene razine vodljivosti. U JFET fototranzistoru, vodljivost kanala kontrolira se naponom na vratima koji stvara zonu osiromašenja. Ova zona osiromašenja smanjuje vodljivost kanala. Kada svjetlost padne na JFET, ona može smanjiti veličinu zone osiromašenja generiranjem dodatnih nositelja naboja u kanalu, čime povećava vodljivost kanala. Manja zona osiromašenja znači veća vodljivost kanala, što povećava struju između izvora i odvoda. [22]

Fototranzistori su vrlo osjetljivi na svjetlost, a njihova struja može se povećati za nekoliko redova veličine kada su izloženi svjetlosti. Brzina s kojom fototranzistor reagira na promjene intenziteta svjetlosti ovisi o njegovoj konstrukciji i materijalima. Fototranzistori se široko koriste u optoelektronici, kao što su detektori svjetlosti, prekidači kontrolirani svjetlom i solarne ćelije. [22].

## 4. LED tehnologija

Svjetleće diode ili LED (*engl. Light Emitting Diode*) su tip diode kojima je glavna zadaća pretvorba iz električne energije u elektromagnetsko zračenje, odnosno svjetlo. LED diode zapravo imaju istu funkciju kao i obične žarulje, međutim značajno su energetski učinkovitije. Njihova povijest započinje početkom 20. stoljeća kada su znanstvenici počeli istraživati poluvodičke materijale i njihovu sposobnost emitiranja svjetla. No, iskorak se događa tek nekoliko desetljeća poslije.

Prvi koraci prema LED tehnologiji napravljeni su 1907. godine kada je britanski znanstvenik H. J. Round otkrio da neki materijali emitiraju svjetlo kada se kroz njih propusti električna struja. Takva pojava naziva se elektroluminiscencija, te je ona osnova rada LED dioda.

Pravi napredak u razvoju LED-ica dogodio se 1960-ih godina. Tijekom tog razdoblja, znanstvenici su razvijali teorije o poluvodičkim materijalima i utvrdili na koji način se emitira svjetlost kroz poluvodiče. U SAD-u, 1962. godine, razvijena je prva LED dioda koja je proizvela svjetlost u vidljivom spektru, crvene boje. Korišten je galij arsenid fosfid (GaAsP) kao poluvodički materijal. Tek kasnije, 1970-ih i 1980-ih, uspjelo se proizvesti LED diode u zelenoj, žutoj i narančastoj boji, što je omogućilo njihovu široku primjenu kao indikatorske lampice u zaslonima i elektroničkim uređajima.

Prvi veliki proboj u LED rasvjeti došao je s razvojem plavih LED dioda. Plave LED diode razvile su se tek u ranim 1990-im godinama. Koristeći galij-nitrid (GaN) kao osnovni materijal, omogućeno je LED tehnologiji da emitira svjetlost visoke energije koja se mogla koristiti za stvaranje bijele svjetlosti kombiniranjem plave s fosfornim premazima ili u kombinaciji s crvenim i zelenim LED diodama.

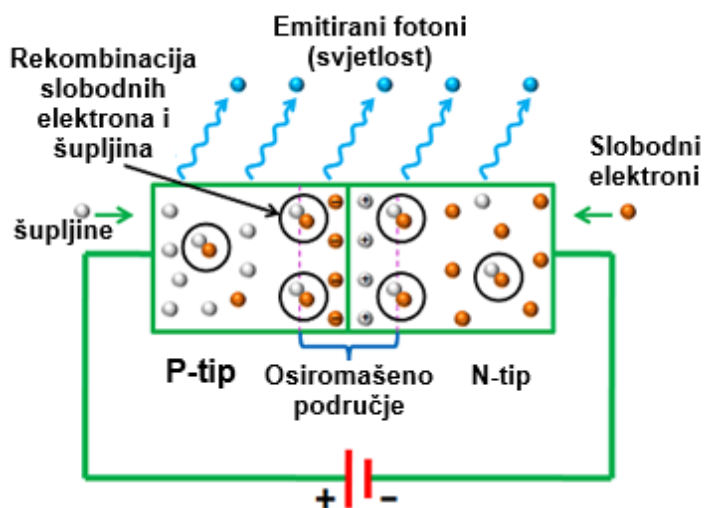
Početkom 2000-ih godina, LED rasvjeta počinje dominirati tržištem, zamjenjujući tradicionalne izvore svjetla poput žarulja sa žarnom niti i fluorescentnih svjetiljki zbog svoje energetske učinkovitosti, dugovječnosti i ekološke prihvatljivosti. Danas se LED koristi na raznim područjima kao što su rasvjeta, zasloni pametnih telefona, televizori, semafori, automobilska svjetla i mnoštvo drugih uređaja [23].



## 4.1 Princip rada LED diode

LED dioda, ili dioda koja emitira svjetlost, koristi osnovne principe poluvodiča kako bi generirala svjetlost. Da bi se razumio princip rada LED diode, potrebno je znati osnovne karakteristike poluvodiča i njihovu ulogu u LED tehnologiji. Poluvodiči su materijali s promjenjivom vodljivošću. U svojoj čistoj formi, poluvodič poput silicija ima slabu vodljivost. Međutim, njihova električna svojstva mogu se poboljšati dodavanjem nečistoća. Kada poluvodič ima dodatne elektrone, poznat je kao poluvodič N-tipa jer ima dodatne negativno nabijene čestice. Kada u poluvodiču ima dodatnih šupljina, on je poznat kao poluvodič P-tipa jer ima dodatne pozitivno nabijene čestice.

Osnovna konstrukcija diode sastoji se od dijela materijala N-tipa i P-tipa spojenog zajedno s elektrodama na svakom kraju. Bez dodavanja vanjskog napona, stvara se osiromašena zona između materijala tipa P i N. Ta zona se stvara zbog toga što se većinski nositelji naboja, elektroni u N-tipu i šupljine u P-tipu poluvodiča, premještaju na mjesto manje koncentracije. Obzirom da su razlike u koncentracijama ogromne, slobodni elektroni kreću se prema P-tipu poluvodiča, a šupljine prema N-tipu. Taj proces naziva se difuzija. Kada elektron s N strane prijeđe u P-stranu, unutar vrlo kratkog puta postoji velika mogućnost da će naletjeti na šupljinu i rekombinirati se s njom, pa će tako prestati postojati jedna šupljina i jedan elektron. Slika 9 prikazuje događanja unutar PN spoja nakon što se propusno polarizira.



Slika 9. Propusno polariziran PN spoj

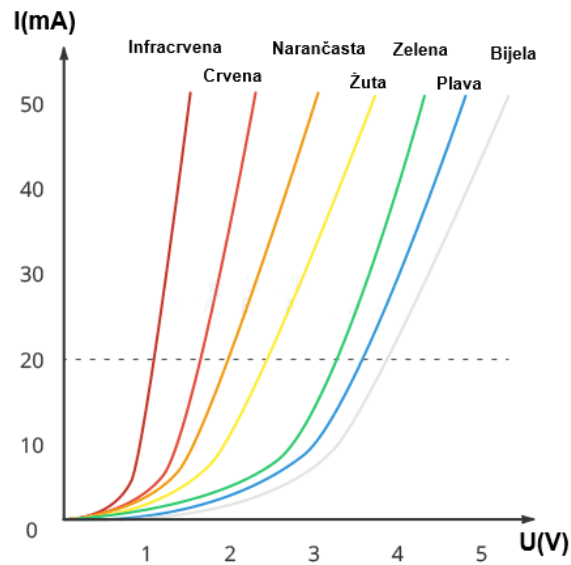
Što je difuzija veća više elektrona i šupljina se rekombinira i osiromašeno područje se širi. Kako osiromašeno područje raste, rezultat je jačanje polja kontakta koje zaustavlja difuziju, sve dok se na kraju difuzija potpuno ne zaustavi i dolazi do ravnoteže [24].

LED dioda je napravljena od posebnih materijala tako da se prilikom rekombinacije dio energije oslobađa u obliku fotona. Pri rekombinaciji sa šupljinom, elektroni iz vodljivog pojasa prelaze u valentni pojas koji je niže energetske razine. Ovo znači da pri svakoj rekombinaciji elektron gubi energiju, koja se kod običnih dioda pretvara u toplinsku, a kod LED dioda dio elektrona gubi energiju u obliku elektromagnetskog zračenja, odnosno fotona. Razlika u energetskim razinama valentne i vodljive vrpce određenog materijala određuje valnu duljinu elektromagnetskog vala koji se oslobađa, to jest, boju emitirane svjetlosti. Otpuštanje fotona dogodit će se samo u slučaju da je LED propusno polarizirana, to jest, pozitivan pol postavljen na P-tip poluvodiča, a negativan na N-tip, te je doveden dovoljno značajan ulazni napon koji će potaknuti rekombinaciju. Tad će se smanjiti osiromašeno područje i većinski nositelji će se lakše premještati na drugu stranu i rekombinirati. Tad će LED dioda početi svijetliti. U slučaju obrnute polarizacije, koju još nazivamo zaporna polarizacija, širi se osiromašeno područje, pa samim time većinski nositelji ne mogu prijeći na drugu stranu pa se ne mogu rekombinirati i LED dioda neće svijetliti [24].

## **4.2 Osnovne karakteristike**

Prije samog korištenja LED dioda, potrebno je poznavati nekoliko njihovih karakteristika kako bi se ispravno upotrijebile. Jedna od najznačajnijih je strujno-naponska karakteristika. Ona prikazuje na koji način će struja koja prolazi kroz LED diodu ovisiti o priključenom naponu. Na U-I karakteristici postoje 3 područja: područje zapiranja, područje vođenja i područje proboja.

Slika 10 prikazuje U-I karakteristike za različite vrste materijala kojima se dobivaju različite boje.



**Slika 10.** U-I Karakteristike za različite LED diode

Sve dok priključeni napon ne dosegne napon praga, LED neće svijetliti. Za različite materijale koji emitiraju različite boje taj napon je drugačiji. U tablici ispod prikazan je napon praga potreban za emitiranje svjetlosti ovisno o materijalu [25].

**Tablica 4.** Napon praga različitih LED dioda [26]

Boja	Napon praga (V)
Crvena	1.6 - 2.0
Žuta	2.0 - 2.2
Narančasta	2.1 - 2.2
Plava	2.4 - 3.7
Zelena	1.9 - 4.0
Ljubičasta	2.7 - 4.0
Ultraljubičasto	3.1 - 4.4
Bijela	3.2 - 3.6

Ako priključeni napon značajno prelazi ove vrijednosti dolazi do proboja zbog toga što iznos struje koji prolazi kroz LED diodu nakon napona praga eksponencijalno raste. Prevelik napon uzrokuje prekomjernu struju kroz LED, što može dovesti do pregrijavanja, smanjenja svjetlosne efikasnosti, skraćanja životnog vijeka i trajnog oštećenja diode. Zbog toga je ključno pravilno ograničiti struju kroz LED korištenjem otpornika ili drugih metoda zaštite.

Osim strujno-naponske karakteristike potrebno je spomenuti i polaritet. Polaritet je pokazatelj simetrije elektroničke komponente. Element je simetričan ako neovisno o tome na koju stranu ga okrenuli prilikom priključivanja u strujni krug, on zadrži identičnu ulogu. Primjer takvog elementa je otpornik. Za razliku od njega, LED dioda, zbog PN spoja, nije simetrična, odnosno dopušta protok struje samo u jednom smjeru.

Kod LED diode, pozitivni izvod se naziva anoda, a negativni izvod se naziva katoda. Da bi LED ispravno radio, anoda LED-a mora biti na višem potencijalu od katode jer struja u LED diodi teče od anode prema katodi [25]. Ako LED diodu spojimo obrnuto, neće se dogoditi ništa jer LED neće provoditi struju. Anodni izvod se prepoznaje na način ta je često duži od izvoda katode u klasičnoj LED diodi.

### **4.3 Vrste LED dioda**

Podjela LED dioda može se izvršiti na mnoštvo načina, a za potrebe ovog rada najbitnije su sljedeće podjele [25, 27]:

- Prema vrsti emitirane svjetlosti
- Prema obliku i veličini
- Prema namjeni

Prema vrsti emitirane svjetlosti mogu se podijeliti na:

- Jednobojne LED diode – imaju mogućnost emitiranja samo jedne boje kad su propusno polarizirane i doveden je odgovarajući napon (minimalno napon praga).

- Dvobojne LED diode - Unutar jednog kućišta se nalaze dvije diode, koje su spojene u suprotnom smjeru jedna naprema drugoj. Dioda koja će zasvijetliti ovisi o polarizaciji.
- RGB LED diode – Kombiniraju tri osnovne boje: crvenu (R), zelenu (G) i plavu (B) u jednom pakiranju. Kombiniranjem različitih intenziteta crvene, zelene i plave svjetlosti, RGB LED diode mogu stvoriti širok spektar boja, uključujući bijelu te omogućuju stvaranje do 16 milijuna različitih boja. Zbog toga se koriste u različim područjima kao što su televizori, monitori, LED trake, indikatori u pametnim uređajima i dekorativna rasvjeta.

Nadalje, prema obliku se mogu podijeliti na:

- Standardne LED diode - poznate kao DIP (*engl. Dual In-Line Package*) LED diode, su klasične LED diode koje se najčešće prepoznaju po svom dizajnu. Imaju tvrdo plastično kućište u obliku kapsule s dvije paralelne žice za povezivanje. Boja plastičnog kućišta obično označava boju svjetlosti koju dioda emitira iako ona nije zadužena za boju koju će LED emitirati već vrsta poluvodičkog elementa koji se nalazi u njoj. Iako su danas manje efikasne od novijih tehnologija poput SMD-a, i dalje se koriste u jednostavnijim aplikacijama [27]. Slika 11 prikazuje fizički izgled standardnih LED dioda.



**Slika 11.** Standardne LED diode [25]

- SMD LED diode (*engl. Surface-Mount Device*) - LED diode koje su montirane izravno na površinu tiskanih ploča. One su male, energetski učinkovite i mogu sadržavati više dioda (npr. crvena, zelena i plava) na jednom čipu, što omogućuje stvaranje različitih boja, uključujući bijelu svjetlost. Zbog svoje veličine i fleksibilnosti, često se koriste u modernim

rasvjetnim i elektroničkim uređajima poput LED traka i zaslona. Upravo taj tip LED diode je korišten u praktičnom radu. Proizvode između 60 i 80 lumena po W, što je značajno veći iznos nego za standardne te se zato često koriste u LED trakama [27]. Slika 12 prikazuje fizički izgled SMD diode.



**Slika 12.** SMD dioda [25]

- COB LED diode(engl. *Chip on Board*) – Sadrže mnogo više LED čipova smještenih vrlo blizu na jednoj ploči, što rezultira ravnomjernijim, jačim svjetlosnim izvorom. Zbog ove gustoće čipova, COB diode emitiraju svjetlost kao jedan snažan izvor bez vidljivih pojedinačnih točkica svjetla. Koriste se u područjima gdje je potrebna jaka, ravnomjerna rasvjeta kao što su ulična rasvjeta, reflektori i industrijska okruženja [27]. Slika 13 prikazuje fizički izgled COB diode.



**Slika 13.** COB dioda [25]

Prema namjeni dijele se na:

- Indikatorske LED diode
- Osvjetljavajuće LED diode
- LED diode za optičke komunikacije

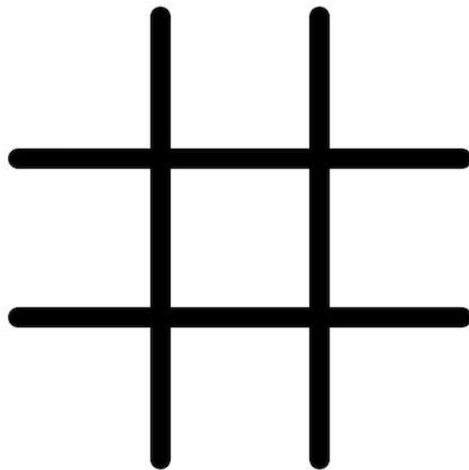
## 4.4 Prednosti korištenja

LED diode možemo smatrati minijaturnim žaruljama koje imaju mnoštvo prednosti u usporedbi sa klasičnim žaruljama sa žarnom niti. Prvenstveno, nemaju niti koje izgaraju, troše značajno manje električne energije i manje se zagrijavaju. Proizvode svjetlost isključivo uz pomoć rekombinacije elektrona i šupljina u poluvodičima te imaju iznimno dug životni vijek. Za razliku od običnih žarulja sa žarnom niti, one mogu svijetliti do nekoliko desetaka puta dulje. Osim toga, znatno su učinkovitije. Žarulje sa žarnom niti u svom procesu proizvodnje svjetla otpuštaju značajnu količinu topline koja je potpuno izgubljena. Kod LED dioda, situacija je u potpunosti obrnuta. Pretvorba iz električne energije u svjetlosnu događa se veoma učinkovito uz puno manje topline kao nusprodukt. LED diode za istu potrošnju energije proizvode više lumena, što znači da daju više vidljive svjetlosti u odnosu na obične žarulje sa žarnom niti. Svjetlosna učinkovitost LED dioda, odnosno sposobnost pretvaranja električne energije u svjetlost, znatno je veća nego kod žarulja sa žarnom niti. Na primjer, žarulja sa žarnom niti od 60 W može proizvesti između 750 i 900 lumena, dok LED žarulja može postići istu razinu svjetline koristeći samo 6 do 8W. LED diode su nešto skuplje u odnosu na žarulje sa žarnom niti, međutim razliku u cijeni nadoknađuju dugotrajnošću. Zbog svih navedenih prednosti, LED diode imaju široku primjenu. Neizostavan su element u elektronici za raznolike uređaje. Neki od primjera su televizori, daljinski upravljači, pametni uređaji i još mnoštvo njih. Osim u elektronici mogu se pronaći i u ostalim područjima poput prometa, medicine i još mnogo njih. Također, koriste se i kao rasvjeta u kućanstvima i industrijskim postrojenjima [28].

## 5. Križić-kružić

Križić-kružić (*engl. Tic-Tac-Toe*), jednostavna je igra koja se igra između dva igrača. Cilj igre je postaviti tri simbola u nizu – bilo horizontalno, vertikalno ili dijagonalno – na 9 polja u 3x3 mreži, to jest matrici. Prvi igrač koji uspije postaviti tri simbola u nizu pobjeđuje, dok se igra smatra izjednačenom ako su svi prostori popunjeni bez pobjednika [29].

Slika 14 prikazuje standardnu matricu 3x3 za igranje igre Križić-kružić.



**Slika 14.** Izgled križić-kružić prostora za igru

Iako točno porijeklo igre nije poznato, slične igre su se igrale tisućama godina. Na primjer, varijacija igre poznata kao "*three men's morris*" bila je popularna u Engleskoj još u 1300. godini, dok se igra "*terni lapilli*" igrala u starom Rimu. Također, Indija je imala svoju verziju igre nazvanu "*Pada*" gdje je cilj bio ostvariti određeni uzorak koristeći ljuske i kamenčiće. Igra se u Engleskoj izvorno nazivala "*noughts and crosses*" (križići i kružići), a prvi put je zabilježeno u znanstvenom časopisu 1858. godine. U 20. stoljeću, igra je preimenovana u "Tic-Tac-Toe" [29, 30].

Igra pripada skupini igara koje se zovu mn-k igre. To su matematičke igre koje se temelje na kombinatorici i teoriji grafova te se igraju na  $m \times n$  površini, odnosno matrici, na kojoj igrači



naizmjenice postavljaju svoje figure (simbole) na ploču slijedeći pravila. Pobjednik je onaj igrač koji uspije postaviti **k** svojih figurica u nizu na ploču [31].

Neke od igara koje imaju isti ili sličan princip su:

- Pente
- Gomoku
- Connect Four
- Misere
- Go

## 5.1 Pravila

Pravila igre križić-kružić su veoma jednostavna. Igrači naizmjenično odabiru prazna polja i upisuju svoj simbol u odabrano polje. Prvi igrač koji započinje igru dobiva simbol križića, odnosno “X”. Drugi igrač upisuje simbol kružića, odnosno “O”. Igra može završiti na 3 načina, a to su:

- pobjeda igrača koji predstavlja križić (X) – ako uspije upisati svoja 3 simbola u nizu, bilo vertikalno, horizontalno ili dijagonalno
- pobjeda igrača koji predstavlja kružić (O) – vrijedi isto kao i kod prvog igrača
- neriješen ishod – ako sva polja budu ispunjena, a nitko nije uspio ostvariti 3 simbola u nizu, nitko nije pobijedio

## 5.2 Strategije

Ako igrači odigraju optimalno, igra križić-kružić uvijek će završiti neriješeno. Kako bi igrači osigurali što povoljniji ishod u igri križić-kružić, ključno je slijediti određeni redoslijed odabira poteza i prioritizirati poteze prema važnosti. Ovo omogućuje igračima da maksimiziraju svoje šanse za pobjedu ili barem osiguraju neriješen ishod. Slijedi popis strateških poteza s prioritetima koji se trebaju razmatrati [32]:

1. Pobjedi: Ako igrač već ima dva simbola u nizu, odmah mora postaviti treći simbol kako bi ostvario pobjedu. Ovo je najviši prioritet jer direktno vodi do pobjede.

2. Blokiraj: Ako protivnik ima dva simbola u nizu i prijeti pobjedom, igrač mora postaviti svoj simbol kako bi blokirao protivnika. Ovo je drugi najvažniji potez kako bi se spriječila neposredna prijetnja.
3. Dvostruka prijetnja: Nastojte stvoriti situaciju gdje imate dvije različite mogućnosti za pobjedu, odnosno dva neblokirana niza sa po dva simbola. Ovaj potez omogućuje igraču da osigura pobjedu u sljedećem potezu jer protivnik može blokirati samo jednu od mogućnosti.
4. Blokiranje protivničke dvostruke prijetnje: Ako postoji moguća dvostruka prijetnja za protivnika, igrač treba blokirati tu prijetnju.
5. Središte: Ako je središnje polje slobodno, igrač ga treba zauzeti kako bi si osigurao najbolji ishod.
6. Suprotni Kut: Ako je protivnik zauzeo kut, igrači bi trebali zauzeti suprotni kut kako bi maksimizirali svoje šanse za pobjedu.
7. Prazni Kut: Ako su svi kutovi slobodni, igrač treba zauzeti jedan od kutova.
8. Prazna Strana: Ako su svi kutovi zauzeti, igrač bi trebao igrati na jednom od srednjih polja na rubovima.

Prvi igrač, iako na početku ima 9 dostupnih polja za igru, zapravo za odabir postoje 3 strateški jedinstvene pozicije. Te pozicije mogu se nazvati kut, strana i sredina [33]. Nadalje će se na nekoliko primjera objasniti potencijalne mogućnosti ovisno o tome koje polje je prvi igrač odabrao za svoj početni potez. Zbog jednostavnosti, na slikama koje će slikovno prikazivati primjere “X” će označavati prvog igrača, a “O” drugog igrača, a  $X_{1,2,3}$  i  $O_{1,2,3}$  označavaju redosljed poteza igrača.

### 5.2.1 $X_1$ u kutu

Ovaj pristup je najpopularniji te pruža najviše različitih mogućnosti za pobjedu prvog igrača. Ako drugi igrač ne odabire za svoj potez središnje polje, “X” može pobijediti na 7 načina [33]. Slika 15 prikazuje sve moguće načine kako “X” može pobijediti ako je  $X_1$  kut, a  $O_1$  nije u sredini.

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>																																				
<table style="width: 100%; border-collapse: collapse;"> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_1</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_1</math></td><td style="padding: 2px 5px;"></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_2</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_3</math></td><td style="padding: 2px 5px;"></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_2</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"></td></tr> </table>	$x_1$	$o_1$		$o_2$	$x_3$		$x_2$			<table style="width: 100%; border-collapse: collapse;"> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_1</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"><math>o_1</math></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_2</math></td><td style="padding: 2px 5px;"></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_3</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"><math>x_2</math></td></tr> </table>	$x_1$		$o_1$		$o_2$		$x_3$		$x_2$	<table style="width: 100%; border-collapse: collapse;"> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_1</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_2</math></td><td style="padding: 2px 5px;"><math>x_2</math></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_1</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"><math>x_3</math></td></tr> </table>	$x_1$	$o_2$	$x_2$	$o_1$					$x_3$	<table style="width: 100%; border-collapse: collapse;"> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_1</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_2</math></td><td style="padding: 2px 5px;"><math>x_2</math></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"><math>o_1</math></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_3</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"></td></tr> </table>	$x_1$	$o_2$	$x_2$			$o_1$	$x_3$		
$x_1$	$o_1$																																						
$o_2$	$x_3$																																						
$x_2$																																							
$x_1$		$o_1$																																					
	$o_2$																																						
$x_3$		$x_2$																																					
$x_1$	$o_2$	$x_2$																																					
$o_1$																																							
		$x_3$																																					
$x_1$	$o_2$	$x_2$																																					
		$o_1$																																					
$x_3$																																							
	<table style="width: 100%; border-collapse: collapse;"> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_1</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_2</math></td><td style="padding: 2px 5px;"><math>x_2</math></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_1</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"><math>x_3</math></td></tr> </table>	$x_1$	$o_2$	$x_2$				$o_1$		$x_3$	<table style="width: 100%; border-collapse: collapse;"> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_1</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_2</math></td><td style="padding: 2px 5px;"><math>x_2</math></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_3</math></td><td style="padding: 2px 5px;"></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_1</math></td><td style="padding: 2px 5px;"></td></tr> </table>	$x_1$	$o_2$	$x_2$		$x_3$			$o_1$		<table style="width: 100%; border-collapse: collapse;"> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_1</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"><math>x_3</math></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_2</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_2</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"><math>o_1</math></td></tr> </table>	$x_1$		$x_3$	$o_2$			$x_2$		$o_1$									
$x_1$	$o_2$	$x_2$																																					
$o_1$		$x_3$																																					
$x_1$	$o_2$	$x_2$																																					
	$x_3$																																						
	$o_1$																																						
$x_1$		$x_3$																																					
$o_2$																																							
$x_2$		$o_1$																																					
	<b>5</b>	<b>6</b>	<b>7</b>																																				

**Slika 15.** Karakteristične situacije 1

Na slici se može primijetiti kako u svakom primjeru “X” stvara dvostruke prijetnje na svom trećem potezu koje će u idućem potezu rezultirati pobjedom jer “O” ne može spriječiti obje prijetnje. Također, nakon što,  $O_1$  nije postavljen u središnje polje, potez  $O_2$  je nužan jer blokira pobjedu za “X”.

Ako ipak “O” za svoj prvi potez odluči odabrati sredinu, tad “X” nema zagarantiranu pobjedu. Slika 16 prikazuje preostale situacije u kojima igrač “X” može pobijediti nakon što je za  $O_1$  odabrano središnje polje.

<table style="width: 100%; border-collapse: collapse;"> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_1</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_2</math></td><td style="padding: 2px 5px;"></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_1</math></td><td style="padding: 2px 5px;"></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_3</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"><math>x_2</math></td></tr> </table>	$x_1$	$o_2$			$o_1$		$x_3$		$x_2$	<table style="width: 100%; border-collapse: collapse;"> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_1</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"><math>x_3</math></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_1</math></td><td style="padding: 2px 5px;"><math>x_2</math></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_2</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="padding: 2px 5px;"></td></tr> </table>	$x_1$		$x_3$		$o_1$	$x_2$	$o_2$			<table style="width: 100%; border-collapse: collapse;"> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_1</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_2</math></td><td style="padding: 2px 5px;"></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>o_1</math></td><td style="padding: 2px 5px;"></td></tr> <tr><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_3</math></td><td style="border-right: 1px solid black; padding: 2px 5px;"><math>x_2</math></td><td style="padding: 2px 5px;"></td></tr> </table>	$x_1$	$o_2$			$o_1$		$x_3$	$x_2$	
$x_1$	$o_2$																												
	$o_1$																												
$x_3$		$x_2$																											
$x_1$		$x_3$																											
	$o_1$	$x_2$																											
$o_2$																													
$x_1$	$o_2$																												
	$o_1$																												
$x_3$	$x_2$																												

**Slika 16.** Karakteristične situacije 2

Kao što slika prikazuje, “X” će pobijediti samo u slučaju da svoj drugi potez odigra u suprotnom kutu od  $X_1$ , a potez  $O_2$  bude postavljen na neka od preostalih polja koja se nalaze u kutu. Međutim,

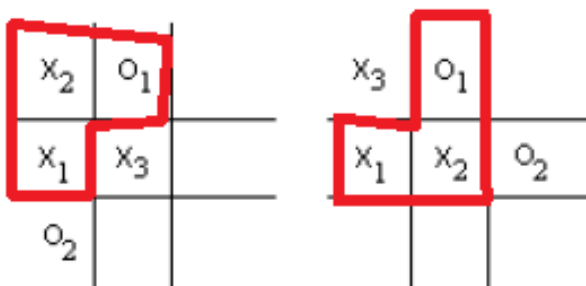
ako "O" odigra optimalno i ne odigra svoj drugi potez u kutu, nego na nekoj od strana, osigurat će si neriješen ishod.

### 5.2.2 $X_1$ na stranama

Kad "X" započinje na nekim od strana, broj zagarantiranih pobjeda pada na 2. Ako je  $X_1$  na bočnim stranama,  $O_1$  može biti postavljen na 5 različitih načina. To su [33]:

- Na susjednim poljima na strani
- Na suprotnom polju na strani
- Na susjednim poljima u kutu
- Na suprotnim poljima u kutu
- U sredini

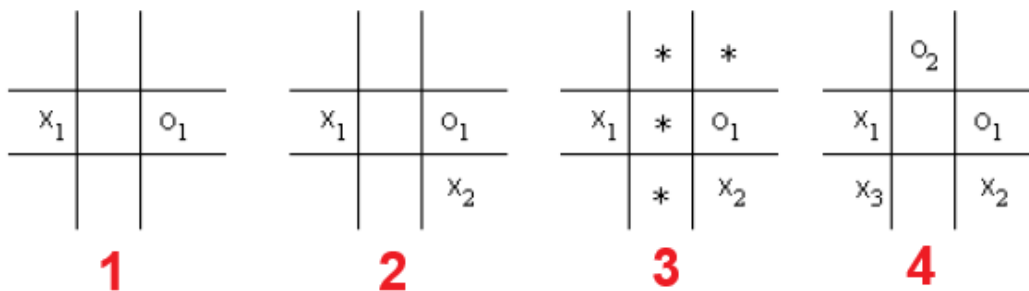
Potrebno je napomenuti da zbog simetričnosti ploče za igru, ova pravila vrijede za sve moguće početne pozicije  $X_1$  na strani (polja 2, 4, 6 i 8). Slika 17 prikazuje zagarantirane pobjede za "X" nakon što je  $O_1$  postavljen na susjednim poljima na strani.



**Slika 17.** Karakteristične situacije 3

Iz slike je vidljivo kako nakon postavljanja  $O_1$  na susjedna polja koji se nalaze na stranama "X" mora svoj sljedeći potez odigrati na način da  $X_2$  postavi odmah uz  $X_1$  i  $O_1$ . Tad će osigurati pobjedu jer će iznuditi sljedeći "O" potez, a nakon toga će "X" moći napraviti dvostruku prijetnju koju "O" neće moći obraniti.

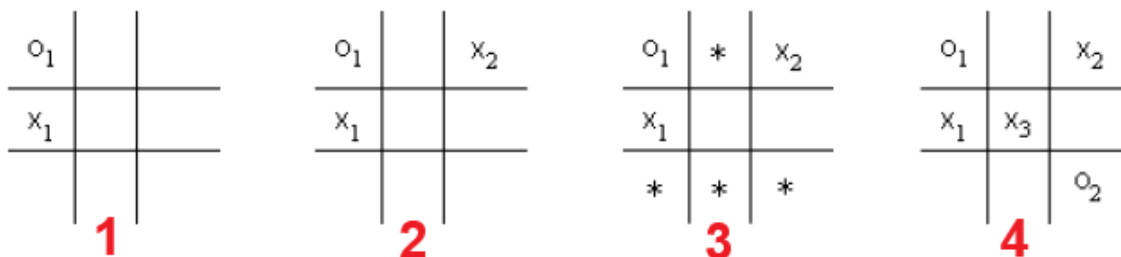
Nadalje, potrebno je razmotriti situaciju kad se  $O_1$  potez odigra na suprotnoj strani od  $X_1$ . Primjer takve situacije prikazan je nizom sličica na slici 18.



**Slika 18.** Karakteristične situacije 4

Prva sličica prikazuje stanje nakon odigranih poteza  $X_1$  i  $O_1$ . Zatim, druga sličica prikazuje strategiju za “X” gdje on mora postaviti  $X_2$  u nekom od kuteva na istoj strani gdje je  $O_1$ . Potom, na sličici 3, polja označena zvjezdicama predstavljaju gubitničke poteze za “O”. Ako  $O_2$  bude odigran na nekim od tih polja, “X” pobjeđuje taktikom dvostruke prijetnje. Jedna od takvih situacija prikazana je na sličici 4. Ako igrač “O” napravi svoj drugi potez na preostalim neoznačenim poljima, osigurat će si neriješen ishod.

Nakon  $O_1$  na suprotnoj strani bit će analizirana situacija kad se  $O_1$  nalazi na susjednim kutnim poljima. Slika 19 prikazuje jedan mogući slijed događanja koji proizlazi iz takve situacije.

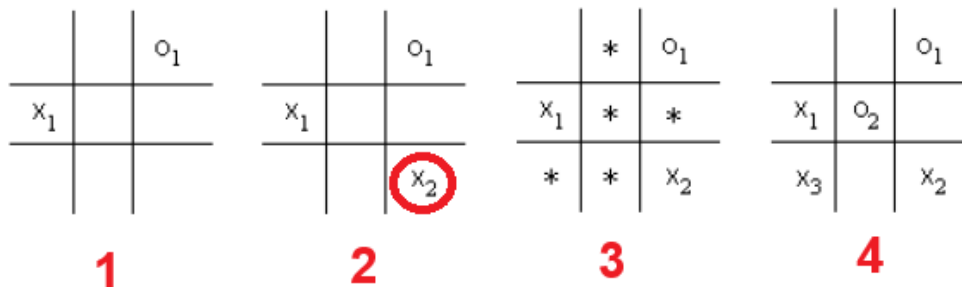


**Slika 19.** Karakteristične situacije 5

Zbog simetričnosti ovdje je također nebitno nalazi li se  $O_1$  s jedne ili druge strane poteza  $X_1$ . Najbolji  $X_2$  potez nalazi se istom retku kao i  $O_1$  što je prikazano na sličici 2. Nakon toga, sličica 3 prikazuje sve gubitničke poteze za “O”, a sličica 4 prikazuje jedan takav ishod. Također, potrebno

je napomenuti da i u ovom slučaju dolazi do pobjede za “X” tek nakon pogreške “O” na drugom potezu.

Osim na susjednim kutevima,  $O_1$  može biti odigran i na suprotnim kutevima. Slika 20 prikazuje karakterističnu situaciju za takav slučaj.



**Slika 20.** Karakteristična situacija 6

Ako je  $O_1$  odigran u suprotnom kutu, tad će “X” imati najveću vjerojatnost za pobjedu ako je njegov sljedeći potez  $X_2$  odigran na drugom preostalom polju koje se nalazi u suprotnom kutu. Taj scenarij je vidljiv na slici 11 pod sličicom 2. Zatim, sličica 3 prikazuje sva gubitnička polja za “O”. Igrač “O” će uspješno odigrati neriješeno samo ako je njegov  $O_2$  potez postavljen u istom retku kao što je  $O_1$ , te u kutu uz potez  $X_1$ . Sličica 4 prikazuje jedan od načina kako “X” može stvoriti dvostruku prijetnju ako “O” ne odabere ispravno polje.

Jedini preostala neobjašnjena situacija kad je  $X_1$  odigran na polju sa strane je kad  $O_1$  zauzme sredinu. U tom slučaju “O” ima najveću vjerojatnost za neriješen ishod, a postoji i mogućnost velike pogreške “X” koja nudi priliku “O” za pobjedu. Slika 21 prikazuje potencijalne mogućnosti “X” za pobjedu.



**Slika 21.** Karakteristične situacije 7

Ako se uzme u obzir da će  $X_2$  biti optimalno postavljen na jedno od 2 prikazana polja, "O" će izgubiti samo ako za svoj  $O_1$  potez odabere polje u suprotnom kutu od  $X_1$  čiji redak već nije ispunjen sa  $X_2$ . Inače, u preostalim potezima  $O_2$  igra će završiti neriješeno.

Osim potencijalnih situacija za pobjedu "X", u ovom slučaju javlja se i mogućnost pogreške "X" igrača. Ta situacija prikazana je na slici 22.

$x_3$		
$x_1$	$o_1$	$x_2$
$o_3$		$o_2$

**Slika 22.** Karakteristična situacija 8

Naime, ako odigra  $X_2$  na suprotnoj strani od  $X_1$ , a između njih je  $O_1$ , tad će uhvatiti sam sebe u zamku i u sljedeća 2 poteza igraču "O" omogućiti stvaranje dvostruke prijetnje.

### 5.2.3 $X_1$ u sredini

Nakon što su detaljno pojašnjenje situacije  $X_1$  u sredini i na strani, preostalo je objasniti primjere kad je  $X_1$  odigran u sredini. Ovaj odabir ostavlja "O" sa dva moguća jedinstvena odabira. Igrač "O" tad može odabrati neko od polja na stranama ili polje koje se nalazi u kutu. Zbog simetričnosti polja za igru, sva polja na stranama smatraju se jednakima, te također sva polja u kutevima imaju isti značaj [33]. Na slici 23, uz slijed sličica, objašnjena je situacija kad je  $O_2$  odigran na strani.

	$o_1$	
	$x_1$	

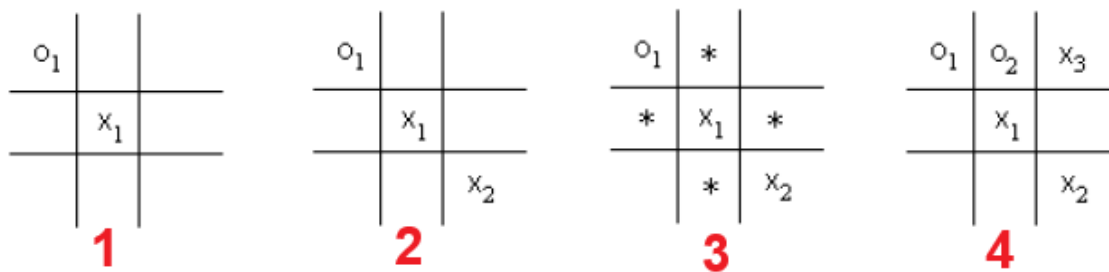
**1**

$x_2$	$o_1$	
$x_3$	$x_1$	
		$o_2$

**2**

**Slika 23.** Karakteristične situacije 9

U tom slučaju “X” ima zagarantiranu pobjedu u četvrtom potezu. Mora postaviti  $X_2$  na bilo koje polje uz  $O_1$ . Potez  $O_2$ , koji blokira dijagonalnu pobjedu za “X”, jedini je preostali potez za “O”. Međutim, to će omogućiti “X” igraču da osigura pobjedu pomoću dvostruke prijetnje u potezu  $X_3$ . Jedina preostala situacija je kad se  $O_1$  nalazi u nekom polju smještenom u kutu. Tad se događa tijekom igre prikazan Slikom 24.



**Slika 24.** Karakteristična situacija 10

Prva sličica u nizu prikazuje početno stanje nakon poteza  $X_1$  i  $O_1$ . Zatim je na sličici 2 prikazan potez  $X_2$  koji ostavlja najviše mogućnosti za pobjedu “X” igrača. To je potez koji je smješten na posljednje preostalo prazno polje u dijagonalnoj liniji koju su stvorili  $X_1$  i  $O_1$ . Nakon toga, sličica 3 prikazuje polja koja “O” mora izbjeći ako ne želi izgubiti. Ako se ipak odluči za neko od tih polja, “X” će pobijediti, a primjer pobjede je prikazan na sličici označenoj brojem 4.



## 6. Praktični dio

U ovom poglavlju obrađeni su praktični aspekti projekta, uključujući sve ključne korake potrebne za realizaciju sustava. Prvo su definirani zahtjevi sustava, zatim opisani korišteni dijelovi, nakon čega slijedi opis procesa izrade. Na kraju je predstavljena implementacija programskog rješenja sustava, čime se zaokružuje cjelokupni praktični dio projekta.

### 6.1 Zahtjevi sustava

Kako bi se uspješno napravila fizička platforma za igru križić-kružić potrebno je u sustav uključiti nekoliko mehanizama. Najprije, potrebno je omogućiti igraču odabir polja na koje želi odigrati svoj potez. Ovo se može izvršiti na nekoliko načina uz pomoć raznih senzora kao što su senzori svjetlosti, senzori pokreta, senzori udaljenosti i slično. Ovaj mehanizam se može napraviti i bez senzora uz korištenje tipkala. Nadalje, potrebno je omogućiti igraču da dobije povratnu informaciju o trenutnom stanju na ploči. To se može izvršiti na mnoštvo načina a neki uključuju LED diodu, LCD ekran i zvučnik. Osim što igrač mora imati informacije o trenutnom stanju i promjeni stanja igre, a u svrhu automatizacije, sustav bi trebao imati upravljačku jedinicu koja će kontrolirati tijek igre i donositi zaključke o ishodima. Za takvo nešto najčešće se koriste mikrokontroleri. Za kraj, sve elemente sustava potrebno je na neki način zaštititi kako ne bi došlo do oštećenja i prestanka rada.

Sažeto rečeno, prilikom izrade igre bilo je potrebno razmisliti o nekoliko čimbenika, a to su:

- Mehanizam unosa poteza
- Povratna informacija o trenutnom stanju igre
- Upravljačka jedinica
- Zaštita sustava od vanjskih utjecaja

## 6.2 Opis korištenih dijelova

U nastavku su opisani svi korišteni dijelovi sustava, zajedno s njihovom funkcijom i ulogom u realizaciji projekta.

Okvir – sastoji se od dva dijela. To je drvena konstrukcija podijeljena na 9 jednakih područja gdje su smješteni ostali elementi te akrilno staklo (pleksiglas) kao pregradni dio.

Arduino Mega 2560 razvojna ploča – detaljno opisana u poglavlju 2.1, koristi se za kontrolu tijeka igre, to jest, za izvršavanje programske logike.

RGB LED WS2812B trake – korištene za osvjetljenje odabranih polja u igri križić-kružić. Kao što je spomenuto u prethodnom poglavlju, one su zadužene za davanje povratnih informacija igračima. Za svako polje korišteno je 24 RGB LED-ica spojenih u nizu. Za ispravan rad potrebno je dovesti istosmjerni napon iznosa 5V. Slika 25 prikazuje fizički izgled RGB LED WS2812B trake.



**Slika 25.** Izgled RGB LED WS2812B trake

LM393 fotoosjetljivi modul – koristi fotootpornik kao glavni element koji očitava razinu svjetlosti u okolini. Za njegov rad potrebno je dovesti istosmjerni napon od 5V na Vcc pin. Za potrebe ovog projekta također je korišten A0 pin za izlazni analogni signal. Slika 26 prikazuje fizički izgled LM393 fotoosjetljivog modula.



**Slika 26.** Izgled LM393 fotoosjetljivog modula [34]

Telefonska žica promjera  $0.6\text{mm}^2$  – za povezivanje elemenata unutar sustava.

Spojnice za žice – za povezivanje različitih vodiča.

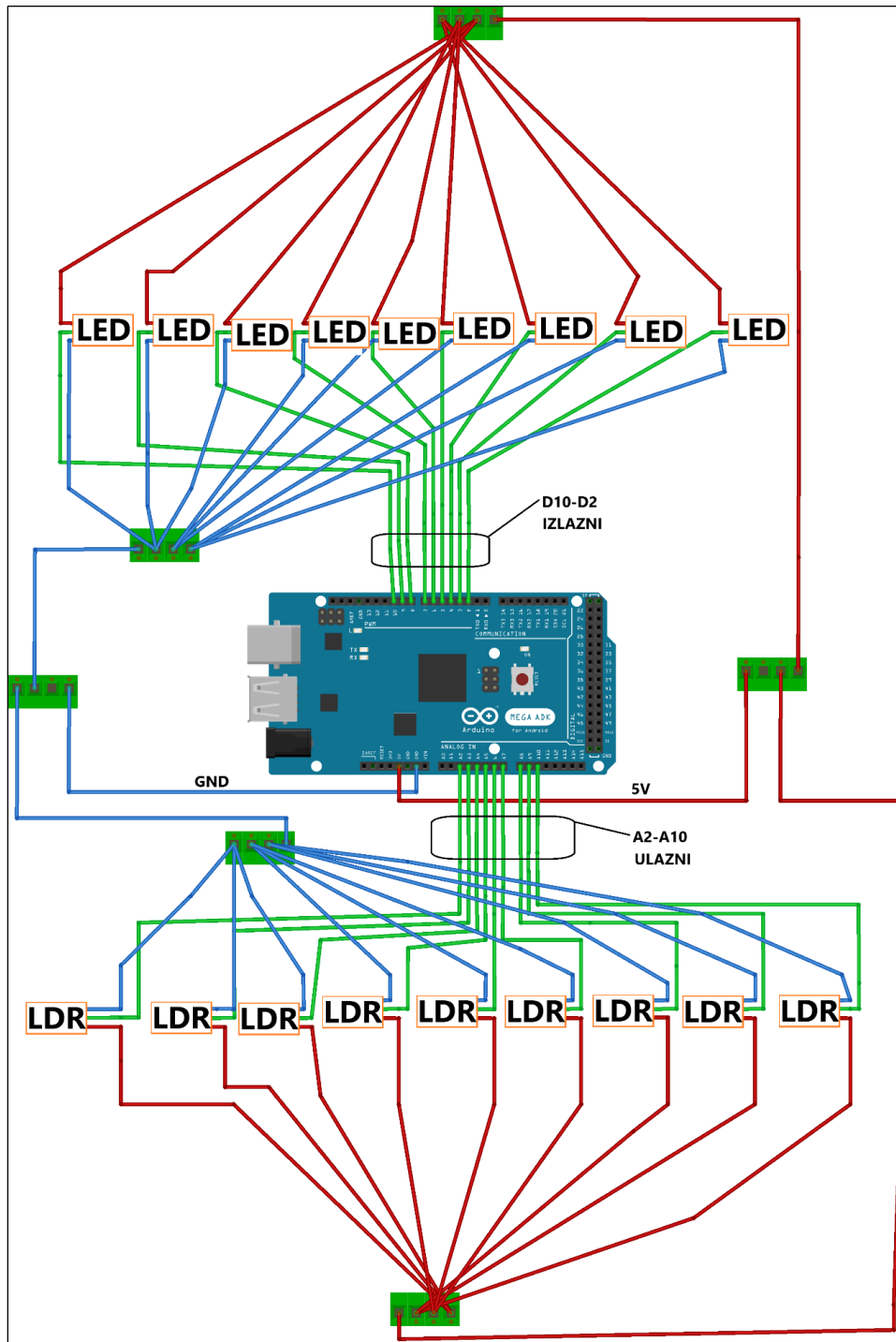
USB-A u USB-B kabel duljine 5m – kao podatkovni kabel za prijenos programa sa Arduino IDE na pločicu.

Osim dijelova korištenih za realizaciju igre potrebno je navesti i korištene alate, a to su:

- Lemilica i lem – za spajanje vodiča na RGB LED trake i LM393 fotoosjetljive module
- Akumulatorska bušilica/odvijač – za kreiranje otvora u okviru gdje će se moći provući vodiči, te za pritezanje vijaka
- Pištolj za vruće lijepljenje – za pričvršćivanje elemenata na okvir
- Vijci – za izradu okvira te pričvršćivanje Arduino pločice na podlogu

### 6.3 Proces izrade

Kako bi se bolje razumjela funkcionalnost sustava, u nastavku na Slici 27 prikazana je detaljna shema koja ilustrira ključne komponente i njihove međusobne odnose.



Slika 27. Shema povezivanja elemenata sustava

Na shemi je vidljivo kako je Arduino mega pločica centralni dio sustava. Na nju su spojeni svi ostali elementi. Nadalje, shema prikazuje 9 fotoosjetljivih modula označenih sa LDR te 9 RGB LED traka, po jedna za svaki senzor koji će očitavati razinu svjetlosti. Na shemi su crvenom bojom označeni svi vodiči na koje se dovodi napajanje od 5V, dok su plavom bojom označeni vodiči za uzemljenje. Obzirom da je broj elemenata u sustavu značajan, a broj pinova za napajanje i uzemljenje na Arduino ploči ograničen, potrebno je točke istog potencijala povezati na neki drugi način. Shema prikazuje povezivanje uz pomoć spojnice. Na shemi je također prikazana dodatna žica zelene koja se povezuje sa svakim senzorom i RGB LED-icom, ali ima različite funkcije ovisno o komponenti. Na senzorima, ta žica je spojena na A0 pin, što označava analogni ulaz preko kojeg senzor šalje signal na Arduino na daljnju obradu. Na RGB LED-icama, ista žica je spojena na D pin, što omogućava slanje podataka (*engl. Data*), preko Arduinovih izlaznih digitalnih pinova. Nakon što je prikazana shema sustava i razjašnjene njegove ključne komponente, slijedi prikaz koraka izrade:

1. Izrada okvira – za potrebe ovog projekta izrađen je drveni okvir dimenzija 750x750mm u kojem će se nalaziti svi elementi sustava. Svako od polja dimenzija je 225x225mm i omogućuje igraču diskretan odabir polja koje želi odigrati. Kako bi igra bila vizualno privlačnija te elementi ostali zaštićeni unutar okvira,
2. Lemljenje vodiča na senzore i RGB LED trake - obzirom na način izrade pojedinih elemenata, bilo je potrebno zalemiti jednu stranu vodiča na njih kako bi se osigurala stabilna i pouzdana električna veza.
3. Pričvršćivanje elemenata unutar okvira – Zbog dimenzija polja ovaj korak je napravljen poslije lemljenja. Za pričvršćivanje elemenata korišten je pištolj za vruće lijepljenje. Vruće ljepilo se brzo suši i osigurava čvrstu vezu, omogućujući brzo i učinkovito spajanje elemenata sa podlogom. Za učvršćivanje Arduino ploče korišteni su vijci koji su pritegnuti akumulatorskim odvijačem.
4. Spajanje vodiča na spojnice i Arduino ploču – Nakon što su svi elementi učvršćeni na podlogu moguće je spojiti drugi dio vodiča zalemljenih u koraku 2. U ovom koraku se spajaju svi izvori napajanja elemenata i uzemljenja u odgovarajuće spojnice te na kraju na pinove Arduinoa. Također, spajaju se i vodiči koji predstavljaju analogni ulaz od senzora na analogne pinove Arduinoa, te D pinovi RGB LED traka na digitalne pinove Arduinoa.

5. Priključivanje na vanjsko napajanje – Nakon što su svi elementi ispravno povezani koristi se USB-A u USB-B kabel za povezivanje Arduina sa računalom. Spajanje Arduina sa računalom omogućava podatkovni prijenos te izvor napajanja potreban za rad.
6. Izrada programske podrške – nakon što su svi fizički elementi ispravno spojeni potrebno je napraviti ispravnu programsku podršku za ciljani sustav. Ovaj postupak detaljno je objašnjen u poglavlju 6.4.

Slika 28 prikazuje izgled unutrašnjosti sustava nakon završetka izrade.



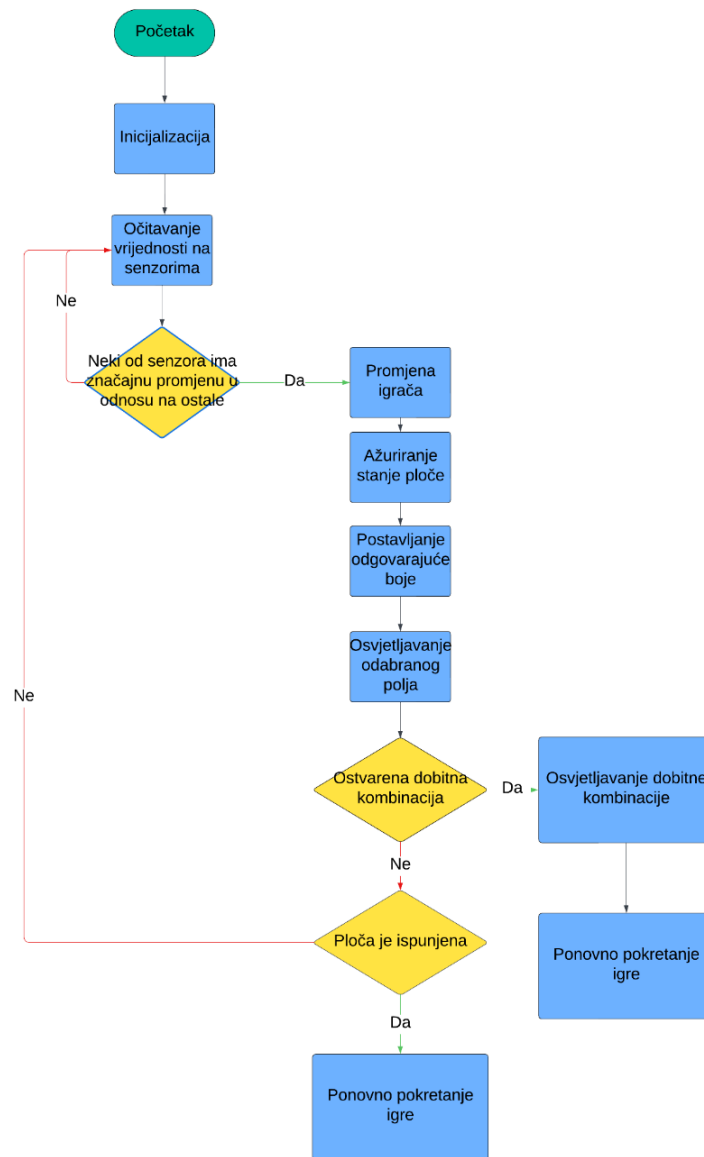
**Slika 28.** Izgled unutrašnjeg dijela sustava

## 6.4 Implementacija

U ovom poglavlju detaljno su opisani ključni dijelovi programskog koda koji omogućuju funkcioniranje sustava te je prikazan dijagram koji ilustrira logički tijek rada programa.

### 6.4.1 Dijagram

Zbog lakšeg praćenja tijeka izvršavanja programskog koda kreiran je dijagram toka. Slika 29 prikazuje dijagram toka izvršavanja igre križić-kružić.



Slika 29. Dijagram toka igre križić-kružić

Dijagram toka govori o ključnim aspektima programa prilikom njegovog izvršavanja. Koraci izvršavanja su sljedeći:

1. Inicijalizacija potrebnih varijabli i definiranje pinova
2. Očitavanje vrijednosti senzora
3. Provjera odstupanja nekog od senzora. U slučaju da nema odstupanja vraćanje na drugi korak
4. Ažuriranje rezultata i osvjetljavanje ploče odgovarajućom bojom
5. Provjera dobitne kombinacije, ako postoji prijeđi na korak 6, inače prijeđi na korak 7
6. Osvjetljavanje dobitne kombinacije i ponovno pokretanje igre
7. Provjera je li ploča ispunjena. Ako je ispunjena igra se ponovno pokreće

Koraci 2 pa do 7 se kontinuirano ponavljaju sve dok igra dobiva vanjski izvor napona.

## 6.4.2 Izvorni kod

U nastavku će detaljno biti prikazan i objašnjen programski kod za igru križić-kružić. Najprije se uključuje biblioteka FastLED, koja se koristi za kontrolu LED traka. Potom, slijedi definiranje konstanti, a njihova uloga je:

- NUM\_FIELDS – označava broj polja u igri križić-kružić
- NUM\_LEDS\_PER\_STRIP – označava broj LED-ica na svakoj traci
- NUM\_STRIPS – predstavlja broj LED traka (po jedna za svako polje igre)
- BRIGHTNESS, LED\_TYPE, COLOR\_ORDER – konfiguriranje LED traka

Prikaz dijela koda za inicijalizaciju potrebni varijabli prikazan je na Slici 30.



```

1  #include <FastLED.h>
2
3  #define NUM_FIELDS 9
4  #define NUM_LEDS_PER_STRIP 24
5  #define NUM_STRIPS 9
6  #define BRIGHTNESS 255
7  #define LED_TYPE WS2811
8  #define COLOR_ORDER RGB
9
10 int fields[NUM_FIELDS] = { A2, A3, A4, A5, A6, A7, A8, A9, A10 };
11 int initialValues[NUM_FIELDS];
12 int gameState[NUM_FIELDS];
13 CRGB leds[NUM_STRIPS][NUM_LEDS_PER_STRIP];
14 int activationCount = 0;
15 const int winCombinations[8][3] = {
16     { 0, 1, 2 },
17     { 3, 4, 5 },
18     { 6, 7, 8 },
19     { 0, 3, 6 },
20     { 1, 4, 7 },
21     { 2, 5, 8 },
22     { 0, 4, 8 },
23     { 2, 4, 6 }
24 };
25 void resetAllLeds(CRGB color = CRGB(0, 0, 0));

```

**Slika 30.** Izvorni kod za inicijalizaciju varijabli

Zatim se definiraju analogni pinovi spremljeni u niz `fields`, inicijaliziraju nizovi za spremanje početnih vrijednosti senzora sa nizom `initialValues` i stanja igre sa nizom `gameState`. Nakon toga, definira se dvodimenzionalni niz `leds` za lakšu kontrolu nad LED trakama. Kako bi se moglo pratiti koji igrač je trenutno na potezu potrebno je inicijalizirati kontrolnu varijablu `activationCount` koja će služiti za postavljanje igrača. Obzirom da je igra križić-kružić definirana određenim pravilima, potrebno je osigurati mogućnost provjere ishoda igre. To se radi definiranjem dvodimenzionalnog niza `winCombinations` koji će služiti za provjeru dobitne kombinacije. Na kraju ovog dijela deklarirana je funkcija `resetAllLed` koja će kasnije biti implementirana.

Slijedi funkcija `setup()`. U njoj se obično postavljaju inicijalne konfiguracije poput definiranja načina rada pinova, pokretanja serijske komunikacije i slično. Ovaj kod će se izvršiti samo jednom. Izvorni kod za naredbe koji će se izvršavati unutar `setup()` funkcije prikazan je na Slici 31.

```

27 void setup() {
28     delay(1500);
29
30     FastLED.addLeds<WS2812B, 2>(leds[0], NUM_LEDS_PER_STRIP);
31     FastLED.addLeds<WS2812B, 3>(leds[1], NUM_LEDS_PER_STRIP);
32     FastLED.addLeds<WS2812B, 4>(leds[2], NUM_LEDS_PER_STRIP);
33     FastLED.addLeds<WS2812B, 5>(leds[3], NUM_LEDS_PER_STRIP);
34     FastLED.addLeds<WS2812B, 6>(leds[4], NUM_LEDS_PER_STRIP);
35     FastLED.addLeds<WS2812B, 7>(leds[5], NUM_LEDS_PER_STRIP);
36     FastLED.addLeds<WS2812B, 8>(leds[6], NUM_LEDS_PER_STRIP);
37     FastLED.addLeds<WS2812B, 9>(leds[7], NUM_LEDS_PER_STRIP);
38     FastLED.addLeds<WS2812B, 10>(leds[8], NUM_LEDS_PER_STRIP);
39
40     // treptanje led traka na početku
41     for (int j = 0; j < NUM_STRIPS; j++) {
42         setStripColor(j, CRGB(255, 0, 255));
43         delay(100);
44         setStripColor(j, CRGB(0, 0, 0));
45         delay(100);
46     }
47
48     // Inicijalizacija input pinova
49     for (int i = 0; i < NUM_FIELDS; i++) {
50         pinMode(fields[i], INPUT);
51         initialValues[i] = analogRead(fields[i]);
52         gameState[i] = 0; // Inicijaliziraj sva polja kao praznih
53     }
54     Serial.begin(9600);
55     delay(500);
56 }

```

**Slika 31.** Izvorni kod funkcije setup

Najprije su uz pomoć funkcije `addLeds` iz biblioteke `FastLED` inicijalizirane LED trake sa definiranjem pripadajućih pinova na Arduino ploči. Potom slijedi probno treptanje LED traka kako bi se utvrdila njihova ispravnost u radu i prikazao vizualno početak rada sustava. Nakon toga, uz pomoć `for` petlje te funkcije `pinMode` definiraju se analogni pinovi na koje su priključeni senzori, očitavaju se početne vrijednosti senzora uz funkciju `analogRead` koje se spremaju u niz `initialValues` te se postavljaju početne vrijednosti polja za igru mijenjajući vrijednosti `gameState` elemenata niza u 0. Za kraj, potrebno je pokrenuti serijsku komunikaciju kako bi se moglo pratiti trenutno stanje programa unutar Arduino IDE.

Nakon `setup()` funkcije dolazi `loop()` funkcija koja je zadužena za kontrolu tijeka izvršavanja programa, Slika 32.

```
58 void loop() {
59     int ledIndex = getMaxChangeIndex();
60
61     if (ledIndex != -1) {
62
63         int player = (activationCount % 2) + 1; // promijeni igrača
64         gameState[ledIndex] = player;          //postavi vrijednost u niz stanja igre ovisno o igraču
65
66         CRGB color = (player == 1) ? CRGB(0, 255, 0) : CRGB(0, 0, 255); // postavi boju ovisno o igraču
67
68         // Osvijetli odabrano polje
69         setStripColor(ledIndex, color);
70
71         // Provjeri postoji li pobjeda
72         int winIndex = checkWin();
73         if (winIndex != -1) {
74             // Treptanje pobjedničke kombinacije i resetiranje igre
75             blinkWinningCombination(winIndex, color, 3, 500);
76             delay(3000);
77             resetGame();
78         } else if (isBoardFull()) {
79             delay(3000);
80             resetGame();
81         }
82
83         activationCount++;
84     }
85
86     delay(2000);
87 }
```

**Slika 32.** Izvorni kod `loop` funkcije

Varijabla `ledIndex` sprema indeks senzora na kojemu je uočena najveća promjena i ta informacija će dalje koristiti za kontrolu igre. Ako se dogodila promjena, potrebno je promijeniti trenutnog igrača, postaviti odgovarajuću vrijednost u niz `gameState` kako bi se ažuriralo trenutno stanje igre. Potom, potrebno je osvijetliti odabrano polje bojom trenutnog igrača. Osvjetljivanje će se izvršiti putem `setStripColor()` funkcije. Nakon što je osvijetljeno odabrano polje potrebno je provjeriti je li trenutni igrač pobijedio. Funkcija `checkWin()` provjerava podudara li se niz `gameState` sa nekim od dobitnih kombinacija definiranih u dvodimenzionalnom nizu `winCombinations`. Ako postoji dobitna kombinacija onda će ona uz pomoć funkcije `blinkWinningCombination()` treperiti nekoliko sekundi, a zatim će se igra ponovno pokrenut uz pomoć funkcije `resetGame()`. Također, ako nema više dostupnih poteza za igru, odnosno ploča je ispunjena, a nitko nije pobijedio, što se provjerava funkcijom `isBoardFull()`, igra će se također ponovno postaviti uz funkciju `resetGame()`. Detaljni

prikaz implementacije svake od navedenih funkcija moguće je vidjeti na poveznici <https://github.com/amrakuzic/diplomskiRad/blob/main/TestSenzoriLed.ino>.

### 6.4.3 Vizualni prikaz sustava

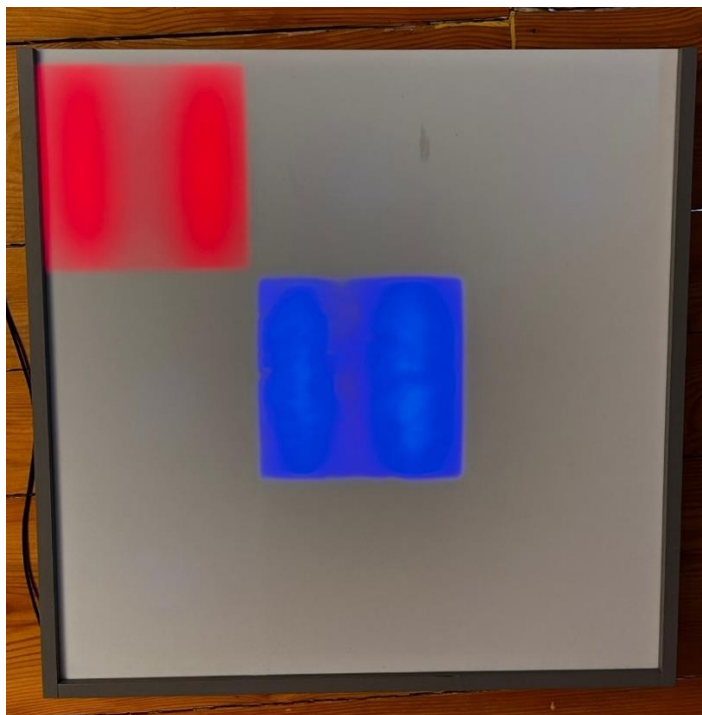
Vizualni prikaz omogućuje pogled na fizički izgled gotovog sustava tijekom igranja igre. Kroz nekoliko slika bit će prikazane karakteristične reakcije sustava u interakciji sa igračima.

Slika 33 prikazuje početak igre dok niti jedan igrač nije odabrao potez. Sustav prepoznaje kako nije potrebno osvjetliti niti jedno polje jer svjetlost podjednako dopire do svih senzora.



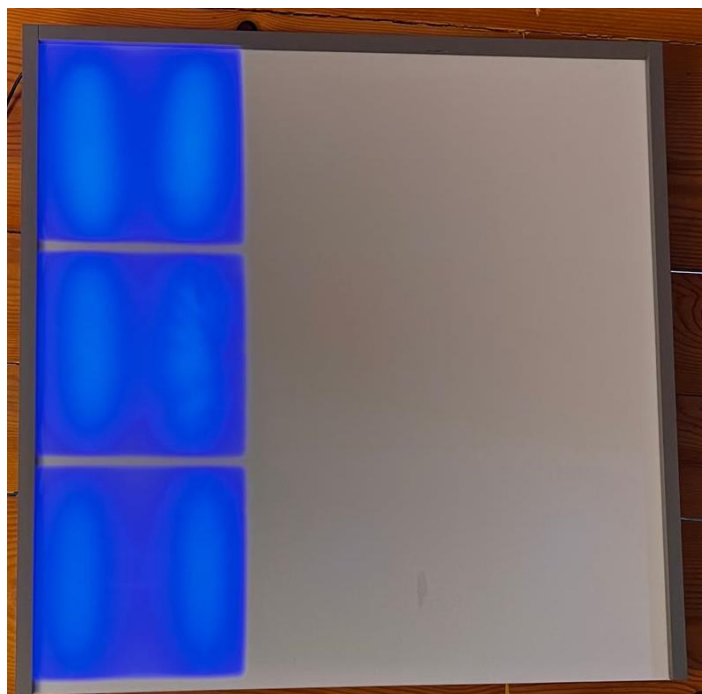
**Slika 33.** Izgled sustava na početku igre

Slika 34 prikazuje fizički izgled igre nakon što su oba igrača odigrala jedan potez. Zbog lakše realizacije umjesto znakova križića i kružića korištene su crvena i plava boja.



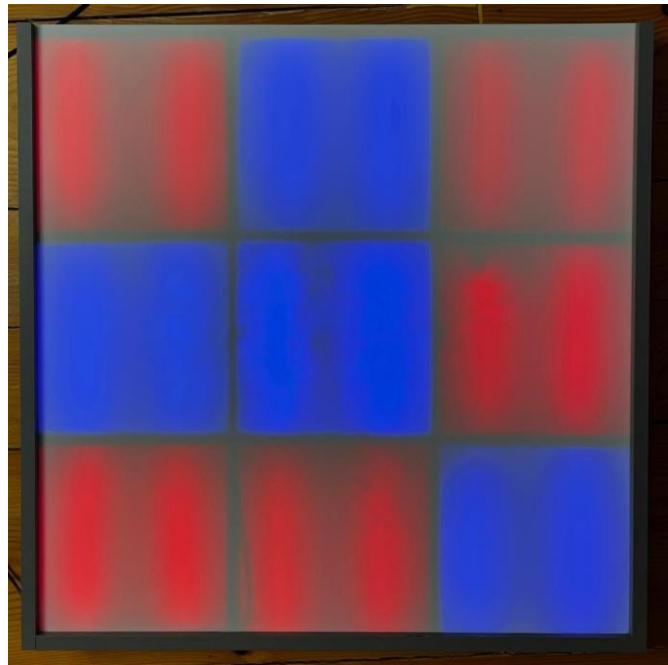
**Slika 34.** Izgled sustava nakon prvih poteza

Slika 35 prikazuje fizički izgled igre u situaciji kad je neki od igrača pobijedio. Pobjednička kombinacija se osvjetljuje, a sva ostala odigrana polja prestaju svijetliti.



**Slika 35.** Izgled sustava nakon ostvarene pobjede

Slika 36 prikazuje fizički izgled nakon što su odigrani svi potezi, a niti jedan igrač nije pobijedio. Nakon ovog ishoda slijedi ponovno pokretane igre.



**Slika 36.** Izgled sustava nakon neriješenog rezultata

## 6.5 Mogućnosti poboljšanja i proširenja funkcionalnosti

Prilikom izrade projekta uočene su mogućnosti poboljšanja sustava te potencijalna proširenja:

Neka od poboljšanja bila bi:

- odabir senzora svjetlosti sa većom osjetljivošću i brzinom odziva
- odabir druge vrste senzora za interakciju igrača sa igrom
- korištenje difuzora za raspršivanje svjetlosti unutar polja

Ovaj sustav mogao bi se proširiti i dobiti neke nove funkcionalnosti, a to bi bile:

- dodavanje AI upravljanog protivnika
- dodavanje mogućnosti igranja preko mreže
- igranje druge igre osim križić-kružić, na primjer igra memorije

## 7. Zaključak

Zaključak ovog rada temelji se na uspješnoj realizaciji sustava za igru križić-kružić korištenjem Arduina, senzora i LED traka. Projekt je započeo analizom zahtjeva sustava i odabirom prikladnih komponenti, što se pokazalo ključnim korakom za uspješnu implementaciju. Arduino platforma, zbog svoje fleksibilnosti, otvorenosti i jednostavnosti programiranja, bila je idealan izbor za realizaciju ovog projekta. Uz mogućnost rada s različitim vrstama senzora i tehnologijama poput LED traka, omogućeno je kreiranje interaktivnog sustava koji korisnicima pruža jasnu i dinamičnu povratnu informaciju o stanju igre.

U središtu ovog sustava nalaze se fotooptički senzorni moduli koji omogućuju korisničku interakciju s igrom. Oni služe kao osjetljivi elementi koji prepoznaju poteze igrača, bilježeći promjene na polju igre križić-kružić. LED trake su ključni vizualni element koji prikazuje status igre, promjene poteza te označava pobjedu ili kraj igre. Upravo kombinacija ovih tehnologija omogućila je stvaranje interaktivnog sustava koji jasno komunicira s korisnicima i pruža ugodno korisničko iskustvo.

Sam proces implementacije obuhvatio je nekoliko faza, od inicijalnog definiranja ciljeva i analize potreba, do praktične izrade i programiranja. Kroz detaljno definiranje zahtjeva igre, bilo je moguće precizno osmisliti način na koji će različiti dijelovi sustava međusobno komunicirati. To je uključivalo planiranje interakcije između senzora, LED traka i Arduina, kao i optimizaciju koda za upravljanje cijelim sustavom. Kroz ovaj proces, pokazalo se koliko je važno temeljito planiranje i razrađivanje ideja prije same implementacije. Na taj način izbjegnute su potencijalne greške, redundantnost u kodu i gubitak vremena.

Iako se radi o jednostavnoj igri, izrada elektroničke verzije igre križić-kružić zahtijevala je pomno planiranje, analizu i implementaciju svih dijelova sustava. Izrada ovog rada pokazala je kako se, kroz korištenje relativno jednostavnih tehnologija, može postići složen i funkcionalan sustav. Ovaj projekt može poslužiti kao temelj za budući razvoj složenijih sustava.

## 8. Literatura

- [1] History of Arduino, [Mrežno]. Dostupno na: <https://circuitprofessor.com/the-history-of-arduino/#before>.
- [2] Sparkfun, What is Arduino?, [Mrežno]. Dostupno na: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>.
- [3] J. Bach, Arduino programming: The Ultimate Beginner's Guide to Learn Arduino Programming step by step, 2020.
- [4] Arduino Applications, [Mrežno]. Dostupno na: <https://trainings.internshala.com/blog/arduino-applications/>
- [5] What is Arduino?, [Mrežno]. Dostupno na: <https://docs.arduino.cc/learn/starting-guide/whats-arduino/>.
- [6] Arduino Mega Picture, [Mrežno]. Dostupno na: <http://www.ulisp.com/show?1LGA>.
- [7] Arduino Specifications, [Mrežno]. Dostupno na: <https://www.electronicshub.org/arduino-mega-pinout/>.
- [8] What is microcontroller?, [Mrežno]. Dostupno na: <https://www.ibm.com/think/topics/microcontroller>.
- [9] W. A. Smith, Ultimate Arduino MEGA 2560 Hardware Manual, 2021.
- [10] R. Turner, ARDUINO PROGRAMMING: THE ULTIMATE BEGINNER'S GUIDE TO LEARN ARDUINO PROGRAMMING STEP BY STEP, 2019.
- [11] Getting started with Arduino, [Mrežno]. Dostupno na: <https://docs.arduino.cc/learn/starting-guide/getting-started-arduino/>.
- [12] Arduino functions, [Mrežno]. Dostupno na: <https://www.arduino.cc/reference/en/>.
- [13] S. M. Paul Scherz, Practical Electronics for Inventors, Fourth Edition.
- [14] J. Fraden, Handbook of Modern Sensors: Physics, Design, and Applications, 2010.



- [15] Automatika principi, [Mrežno]. Dostupno na: <https://www.balluff.com/en-de/basics-of-automation/fundamentals-of-automation/function-construction-technologies-of>.
- [16] I. Petrović, Statičke i dinamičke karakteristike instrumenata, Završni rad, 2023.
- [17] Osnovne karakteristike senzora, [Mrežno]. Dostupno na: <https://ieda.ust.hk/dfaculty/ajay/courses/alp/ieem110/lecs/sensors/sensors.html>.
- [18] N. Ida, Sensors, Actuators, and their Interfaces: A multidisciplinary introduction, 2nd edition, 2020.
- [19] Photoresistor Basics: Types, Principles and Applications,« [Mrežno]. Dostupno na: [https://eclass.uoa.gr/modules/document/file.php/PRIMEDU900/Photoresistor%20Basics\\_%20Types%2C%20Principles%20and%20Applications%20-%20Utmel.pdf](https://eclass.uoa.gr/modules/document/file.php/PRIMEDU900/Photoresistor%20Basics_%20Types%2C%20Principles%20and%20Applications%20-%20Utmel.pdf).
- [20] »Fotodioda prikaz,« [Mrežno]. Dostupno na: <https://www.shutterstock.com/image-vector/vector-electronic-circuit-symbol-photodiode-2183967021>.
- [21] Phototransistor simbols, [Mrežno]. Dostupno na: <https://naradaelectronics.rw/product/phototransistor-ir-receiver-sensor/>.
- [22] Phototransistor basics, [Mrežno]. Dostupno na: <https://www.elprocus.com/phototransistor-basics-circuit-diagram-advantages-applications/>.
- [23] History of Leds, [Mrežno]. Dostupno na: <https://www.shineretrofits.com/lighting-center/lighting-resources/history-of-led-lighting/>.
- [24] How LEDs work?, [Mrežno]. Dostupno na: <https://learn.sparkfun.com/tutorials/light-emitting-diodes-leds/all>.
- [25] Led characteristics, [Mrežno]. Dostupno na: <https://www.electronicshub.org/led-light-emitting-diode/>.
- [26] G. Marko, Svjetleće (LED) diode, 2016.
- [27] Led podjela, [Mrežno]. Dostupno na: <https://www.ledkia.com/blog/uk/types-of-led-diodes-available-on-the-market/>.

- [28] LED diode usage, [Mrežno]. Dostupno na:  
<https://electronics.howstuffworks.com/led.htm>.
- [29] P. Baum, Tic-Tac-Toe, 1975., [Mrežno] Dostupno na:  
[https://www.researchgate.net/publication/316534681\\_Tic-Tac-Toe](https://www.researchgate.net/publication/316534681_Tic-Tac-Toe)
- [30] A. S.Choi, Tic-Tac-Toe, 2020., [Mrežno] Dostupno na: <https://momath.org/wp-content/uploads/2021/08/Alyssa-Choi-Tic-Tac-Toe.pdf>
- [31] M. Shi, Mnk Games, 2017., [Mrežno] Dostupno na:  
<http://simonrs.com/eulercircle/cgt/mnk-games.pdf>
- [32] Tic-tac-toe Strategy, [Mrežno]. Dostupno na: <https://en.wikipedia.org/wiki/Tic-tac-toe>.
- [33] R. Aycock, How to Win at Tic-Tac-Toe, 2002.
- [34] LM393 Photosensitive module, [Mrežno]. Dostupno na: <https://www.flyrobo.in/lm393-photosensitive-ldr-sensor-module>.
- [35] Elektroluminscencija, [Mrežno]. Dostupno na:  
<https://tehnicki.lzmk.hr/clanak/elektroluminiscencija>.
- [36] Led tehnologija [Mrežno]. Dostupno na:  
<https://commercialledlights.com/blog/lighting-articles/led-technology/?srsltid=AfmBOoqYFwTMeTLrpk4WV1honHmln8kzCTIAdho7wk8zCJFZKwpS2DqO> .

## Popis slika

<b>Slika 1.</b> Arduino Mega 2560 razvojna pločica [6] .....	4
<b>Slika 2.</b> ATmega2560 mikrokontroler.....	6
<b>Slika 3.</b> Priključci za napajanje na Arduino MEGA 2560 [9].....	7
<b>Slika 4.</b> Arduino razvojno okruženje.....	11
<b>Slika 5.</b> Prikaz senzora u složenom sustavu [14] .....	15
<b>Slika 6.</b> Simbol fotootpornika (lijevo) i njegov fizički izgled (desno) [19] .....	19
<b>Slika 7.</b> Simbol fotodiode (lijevo) i fizički izgled (desno) [20] .....	20
<b>Slika 8.</b> Simboli BJT I FET fototranzistora (lijevo) i njegov fizički izgled (desno) [21] .....	21
<b>Slika 9.</b> Propusno polariziran PN spoj.....	24
<b>Slika 10.</b> U-I Karakteristike za različite LED diode.....	26
<b>Slika 11.</b> Standardne LED diode [25].....	28
<b>Slika 12.</b> SMD dioda [25].....	29
<b>Slika 13.</b> COB dioda [25] .....	29
<b>Slika 14.</b> Izgled križić-kružić prostora za igru .....	31
<b>Slika 15.</b> Karakteristične situacije 1 .....	34
<b>Slika 16.</b> Karakteristične situacije 2 .....	34
<b>Slika 17.</b> Karakteristične situacije 3 .....	35
<b>Slika 18.</b> Karakteristične situacije 4 .....	36
<b>Slika 19.</b> Karakteristične situacije 5 .....	36
<b>Slika 20.</b> Karakteristična situacija 6 .....	37
<b>Slika 21.</b> Karakteristične situacije 7 .....	37
<b>Slika 22.</b> Karakteristična situacija 8 .....	38
<b>Slika 23.</b> Karakteristične situacije 9 .....	38
<b>Slika 24.</b> Karakteristična situacija 10 .....	39
<b>Slika 25.</b> Izgled RGB LED WS2812B trake .....	41
<b>Slika 26.</b> Izgled LM393 fotoosjetljivog modula [34].....	41
<b>Slika 27.</b> Shema povezivanja elemenata sustava.....	43
<b>Slika 28.</b> Izgled unutrašnjeg dijela sustava.....	45
<b>Slika 29.</b> Dijagram toka igre križić-kružić .....	46
<b>Slika 30.</b> Izvorni kod za inicijalizaciju varijabli.....	48

<b>Slika 31.</b> Izvorni kod funkcije setup.....	49
<b>Slika 32.</b> Izvorni kod loop funkcije .....	50
<b>Slika 33.</b> Izgled sustava na početku igre .....	51
<b>Slika 34.</b> Izgled sustava nakon prvih poteza .....	52
<b>Slika 35.</b> Izgled sustava nakon ostvarene pobjede .....	52
<b>Slika 36.</b> Izgled sustava nakon neriješenog rezultata .....	53

## Popis tablica

<b>Tablica 1.</b> Tehničke specifikacije Arduino Mega 2560 .....	5
<b>Tablica 2.</b> LED indikatori na Arduino Mega 2560 ploči [9] .....	10
<b>Tablica 3.</b> Podjela senzora prema primjeni [18] .....	17
<b>Tablica 4.</b> Napon praga različitih LED dioda [26].....	26