

AKADEMIJA TEHNIČKO-UMETNIČKIH
STRUKOVNIH STUDIJA BEOGRAD
ODSEK VISOKA ŠKOLA
ELEKTROTEHNIKE I RAČUNARSTVA

Amela Zeković

PAMETNI UREĐAJI I KOMUNIKACIJE

prvo izdanje

Beograd, 2025.

dr Amela Zeković

Pametni uređaji i komunikacije

Recenzenti:

dr Ana Gavrovska

dr Slavica Marinković

Obrada i priprema teksta:

dr Amela Zeković

dr Vladimir Cerić

Dizajn korica: struk. mast. inž. Nenad Tolić

Izdavač: Visoka škola elektrotehnike i računarstva

Akademija tehničko-umetničkih strukovnih studija Beograd

Štamparija: BIROGRAF COMP D.O.O. BEOGRAD

Tiraž: 45 primeraka

ISBN: 978-86-6090-185-1

Nastavno-stručno veće Visoke škole elektrotehnike i računarstva strukovnih studija u Beogradu na svojoj sednici odobrilo je izdavanje i korišćenje ovog priručnika u nastavi.

CIP - Каталогизација у публикацији

Народна библиотека Србије, Београд

004.7:004.383(075.8)

ЗЕКОВИЋ, Амела, 1983-

Pametni uređaji i komunikacije / Amela Zeković. - 1. izd. - Beograd : Akademija tehničko-umetničkih strukovnih studija, Visoka škola elektrotehnike i računarstva, 2025 (Beograd : Birograf Comp). - 119 str. : ilustr. ; 25 cm

Tiraž 45. - Bibliografija: str. 113-115.

ISBN 978-86-6090-185-1

a) Паметни уређаји – Мрежно повезивање

COBISS.SR-ID 178673673

Sadržaj

Predgovor	iii
 I Ugrađeni hardver	 1
1 Električna kola i komponente	3
1.1 Uvod u hardver pametnih uređaja	3
1.2 Dizajn električnih kola	4
1.3 Osnovni električni pojmovi	5
1.4 Koraci pri dizajnu električnog kola pametnog uređaja .	6
1.5 Često korišćena namenska električna kola	7
1.5.1 Ograničavač struje otpornikom	7
1.5.2 Promene napona (Delitelj napona, Regulator, Pro- mene nivoa napona)	8
1.5.3 Stabilizator napona	9
1.5.4 Pull-up i pull-down otpornici (Eliminacija šuma pri otvorenoj vezi)	9
1.5.5 Kola sa operacionim pojačavačem	10
1.5.6 Tranzistorski prekidač	11
 2 Električni elementi pametnih uređaja	 13
2.1 Električni elementi pametnih uređaja koji koriste svetlost	13
2.2 Električni elementi pametnih uređaja namenjeni modi- fikaciji napona	15
2.3 Električni elementi pametnih uređaja za pomeranje i/ili kretanje	18
2.4 Električni elementi pametnih uređaja za opservaciju (sen- zori)	20

3	Integrirana kola za pametne uređaje	23
3.1	Integrirana kola	23
3.1.1	Uvod u integrirana kola	23
3.1.2	Često korišćena integrirana kola	25
3.2	Dodatne ploče namenskih uređaja	28
4	Procesori i mikrokontroleri	35
4.1	Uvod	35
4.2	Kako izabrati mikrokontroler?	39
4.3	Najpoznatiji mikrokontroleri	40
5	Programabilna kola	43
5.1	FPGA	43
5.2	ASICs	45
5.3	GPUs	46
5.4	SoC (System on Chip)	47
5.5	Specijalne procesorske jedinice	48
II	Komunikacije	51
6	Hardverska komunikacija upravljačkih računarskih elemenata	53
6.1	Osnove serijske i paralelne komunikacije	53
6.1.1	Serijska komunikacija	53
6.1.2	Paralelna komunikacija	54
6.1.3	Tehnološki trendovi u razvoju serijske i paralelne komunikacije	56
6.1.4	Talasni oblici digitalnih signala	56
6.2	Serijski protokoli za upravljačke računarske sisteme	57
6.2.1	I ² C komunikacija	57
6.2.2	Uvod u I3C	58
6.2.3	SPI	59
7	Pregled komunikacionih sistema kratkog i lokalnog dometa za pametne uređaje	61
7.1	Bluetooth i Bluetooth Low Energy (BLE)	61
7.1.1	Uvod u Bluetooth komunikaciju	61
7.1.2	Bluetooth Low Energy (BLE)	62
7.1.3	Poređenje klasičnog Bluetootha i BLE	63

7.2	ZigBee komunikacija	63
7.3	RFID i NFC — identifikaciona komunikacija	64
7.4	Ethernet i Wi-Fi komunikacija	65
7.4.1	Ethernet — žičana komunikacija velikog kapaciteta	65
7.4.2	Wi-Fi — bežična fleksibilnost za pametne sisteme	66
8	Mreže sa širim dometom i uporedni pregled tehnologija	67
8.1	LoRa — dug domet uz nisku potrošnju	67
8.2	Mobilne telekomunikacije (2G–5G)	68
8.3	NB-IoT — IoT proširenje mobilnih mreža	69
8.4	Cloud i arhitekture za ugrađene uređaje	69
8.5	Uporedni pregled i implikacije za ugrađene uređaje	70
III	Embedded mašinsko učenje	73
9	Ograničenja za mašinsko učenje u namenskim računarskim sistemima	75
9.1	Uvod	76
9.2	Hardverski segment namenskih računarskih sistema	76
9.3	Izazovi u pogledu softvera embedded sistema	78
9.4	Izazovi ML algoritama za TinyML	80
9.4.1	TinyML i namenski procesori	82
9.4.2	Kompresija ML modela	83
9.4.3	Softverski okviri za izvršavanje ML modela	84
10	Osnove mašinskog učenja	87
10.1	Eksplisitno i ML kodovanje	87
10.2	Merenje tačnosti i gradijentni pad	89
10.3	Uvod u neuronske mreže	91
10.4	Vrste slojeva u neuronskim mrežama	92
10.5	Osnovni blokovi neuronskih mreža: inicijalizacija i učenje	94
11	Neuronske mreže	97
11.1	Inicijalizacija i učenje na primeru linearne regresije	97
11.2	Funkcija $f(x)$ sa više od jednog neurona	99
11.3	Uvod u klasifikaciju	101
11.3.1	Skup podataka MNIST	102
11.3.2	Struktura mreže	102
11.3.3	Obuka i evaluacija	103

11.4 Podaci za učenje, validaciju i testiranje	103
11.5 Konvolucione neuronske mreže	104
12 Primene Edge AI i TinyML	107
12.1 Uvod	107
12.2 Uloga senzora u TinyML	108
12.3 Industrijske primene	108
12.4 Primene u zdravstvu i ljudskim aktivnostima	109
12.5 Primene u ekologiji i životinjskom svetu	110
12.6 Obrada audio i vizuelnih signala	111
12.7 Dizajn i odgovornost	111
Literatura	113
Ključne reči	117

Predgovor

Ovaj udžbenik je namenjen studentima prve godine odseka Visoka škola elektrotehnike i računarstva strukovnih studija u Beogradu, za predmet Pametni uređaji i komunikacije. Predmet matično pripada studijskom programu Elektronika i telekomunikacije, ali ga često biraju kao izborni predmet sa drugog studijskog programa i studenti studijskih programa Automatika i sistemi upravljanja vozilima, Računarska tehnika, Nove računarske tehnologije i Informacioni sistemi, zbog veze sadržaja ovog predmeta sa njihovim matičnim studijskim programima. Pored studenata odseka Visoka škola elektrotehnike i računarstva, udžbenik može da bude od koristi i svim drugima koji žele da kroz praktične primere steknu osnove iz ovog predmeta.

Udžbenik sadrži tri celine.

Prva celina se fokusira na ugrađeni hardver u pametnim uređajima. Obrađuje CPU, mikrokontrolere, FPGA, ASIC, GPU i SoC, sa naglaskom na programabilnost, fleksibilnost i praktičnu integraciju u složene sisteme.

Druga celina se fokusira na komunikacione tehnologije za pametne uređaje. Obrađuje karakteristike i principe u slučaju mreža sa malim i većim dometom, energetske efikasnosti, topologije mreža i integraciju sa cloud i edge arhitekturama, sa posebnim naglaskom na optimizaciju za uređaje sa ograničenim resursima.

Treća celina se odnosi na primenu mašinskog učenja u okviru namenskih računarskih (*embedded*) sistema. Obrađuje specifičnosti hardverskih i softverskih ograničenja, optimizaciju modela za uređaje sa ograničenim resursima i praktične pristupe implementaciji TinyML rešenja.

Autor se zahvaljuje recenzentima dr Ani Gavrovskoj i dr Slavici Marinković na korisnim predlozima i sugestijama koji su doprineli kvalitetu ovog udžbenika.

Pored najveće pažnje tokom formiranja i pisanja ovog udžbenika moguće je da su se potkrale greške. Autor će biti zahvalan svim čitaocima koji ukažu na njih ili daju predloge i sugestije.

U Beogradu, oktobar 2025. godine

Autor

Deo I

Ugrađeni hardver

Glava 1

Električna kola i komponente

Cilj poglavlja

Cilj poglavlja je da pruži pregled osnovnih hardverskih komponenti pametnih uređaja, sa fokusom na električne komponente i kola. Objašnjava ulogu hardvera u funkcionisanju pametnih uređaja, povezanost sa softverom i algoritmima, osnove elektrotehnike, dizajn električnih kola i primenu često korišćenih namenskih kola, uključujući zaštitu, regulaciju napona, eliminaciju šuma, pojačavanje signala i tranzistorsku logiku.

1.1 Uvod u hardver pametnih uređaja

Za rad i primenu pametnih uređaja neophodna je kombinacija hardvera i softvera, kao i odgovarajućih pratećih algoritama i protokola. Za dobro razumevanje neophodan je detaljan i povezan pregled fizičkih komponenti koje čine ovaj sistem.

Priprema za samu primenu podrazumeva i analizu koje su to ključne i prateće komponente neophodne za razvoj određenog pametnog uređaja. Na primer, ako se radi na pametnom dronu koji treba da radi analizu poljoprivrednih uslova, potrebno je da se razmotri koje komponente su neophodne u pogledu senzora, a koje za eventualni autonomni rad. Drugi primer je detekcija anomalije u radu mašine, na primer na osnovu vibracija i anomalija u odnosu na uobičajeno ponašanje, u različitim mehaničkim ili industrijskim sistemima. Koji senzori bi bili potrebni pre same analize za mašinskim učenjem? U primeru pametnog sata sa

određenih namena, potrebno je razmotriti koji senzori su potrebni, ali kojih dimenzija i koje potrošnje energije. Sledeći primer je analiza ponašanja kupca u prodavnicama, na primer u pogledu praćenja pogleda i zadržavanja pogleda. Koji zaključci mogu da se izvuku? Ovo su neki od važnih naprednijih primena čiji segmenti mogu da budu korišćeni u navedenim slučajevima, ali i u mnogim drugim.

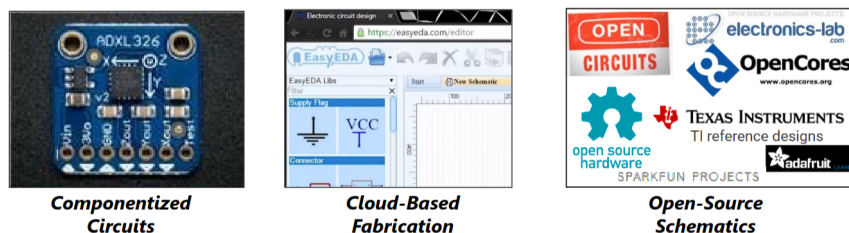
U pozadini svih ovih primena nalazi se izabrani odgovarajući hardver, sa električnim kolima i komponentama koje omogućavaju njegovo postojanje i funkcionisanje. Na sam hardver se dalje dodaju odgovarajući softverski elementi, kao što su programiranje namenske računarske komponente pametnog sistema i algoritmi mašinskog učenja u slučaju ovakve naprednije primene.

1.2 Dizajn električnih kola

Razvoj i složenost u dizajnu električnih kola menjala se tokom vremena. Naravno, uvek važi da sjajno razumevanje rada električnog kola i mogućnost kompletnog dizajna i analize od početka do svih detalja nudi opcije za vladanje svakim segmentom njegovog rada i svih mogućih izmena. Ovo naravno zahteva veliko znanje, vreme i novac. Takođe, kvalitet izrade je takođe nešto o čemu je fizički potrebno voditi računa.

Međutim, postepeno tokom godina razvijen je veliki broj drugih mogućnosti u dizajnu i razvoju električnih kola, posebno u poslednje vreme. Na primer, postoje opcije modularnog razvoja i korišćenje dobro razvijenih postojećih rešenja za kola određenih namena prilikom razvoja celog uređaja. Postoji veliki broj često korišćenih električnih kola potrebnih prilikom razvoja i nije neophodan njihov razvoj od nule. Dalje, moguće je korišćenje i upakovanih integrisanih kola za rešavanje različitih segmenata uređaja. U ovom slučaju čak nije neophodno ni zalaziti u detalje unutrašnjeg rešenja, ono je već optimizovano i upakovano u integrisano kolo.

Zatim, razvijeni su i softveri za kreiranje prototipa, dizajn i analizu kola, pa i dalje za kreiranje kompletnih štampanih ploča, koje je moguće, čak i direktno iz softvera, poslati firmama koje se bave njihovom fabrikacijom i štampanjem na pouzdan način, a da dalje izrađena ploča stigne na željenu adresu. Postupak štampe se obavlja u sistemima sa komercijalnim kvalitetom i pogonima, što ne bi moglo da se uradi



Slika 1.1: Evolucija dizajna električnih kola.

na malom nivou i bez namenskih firmi. Na ovaj način je moguće vrlo stabilno koristiti rigorozne sisteme koji primenjuju ove tehnologije.

Poslednji koraci u razvoju su dizajn električnih kola, prateći slobodne softvere, slobodne programske jezike i njihove biblioteke (Python), slobodne operativne sisteme (kao što je Linux), kao i postojanje slobodnog otvorenog hardvera. Na ovaj način, moguće je korišćenje postojećih rešenja, njihova modifikacija i usavršavanje, pa čak i IP jezgra procesora i njihovo spuštanje na hardverski element – čip.

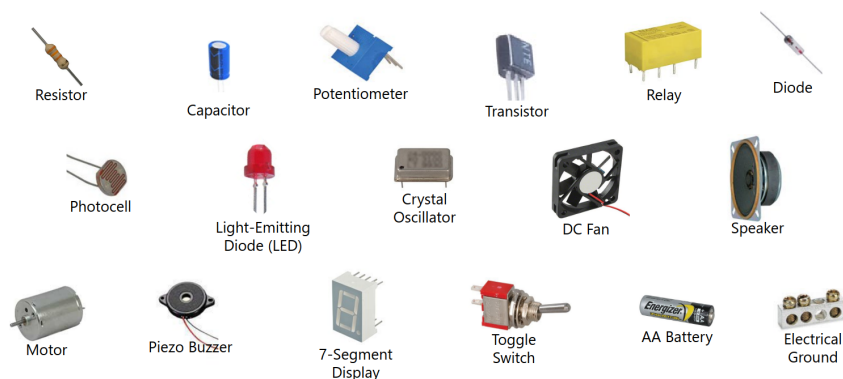
Ovi različiti segmenti omogućavaju razvoj veoma naprednih pametnih uređaja, a da se ne utroše ogromne količine vremena ili novca za izgradnju svih sitnih detalja. Naravno, za spajanje celina, kombinovanje i prepoznavanje potrebnih segmenata, neophodno je razumevanje elemenata i kola.

1.3 Osnovni električni pojmovi

Struja podrazumeva usmereno kretanje valentnih elektrona kroz provodnik. Različite komponente se različito ponašaju u odnosu na struju. Na primer, otpornik smanjuje protok elektrona, a kondenzator skladišti naelektrisanje. Otpornici i kondenzatori mogu postojati i kao komponente, ali njihove osobine često moraju da se uzmu u obzir i kod drugih komponenti kod kojih se ne vide direktno.

Postoji veoma veliki broj različitih komponenti. Za uspešno razumevanje, dizajn i razvoj pametnih uređaja neophodno je poznavanje najčešće korišćenih. Prilikom dizajna kola crtaju se električne šeme, gde se koriste simboli električnih komponenti i vrši njihovo povezivanje. Ovo omogućava znatno pregledniji i precizniji prikaz.

Na primer, otpornik koristi simbol koji asocira na smanjenje struje, a kondenzator simbol koji podseća na paralelne ploče gde se sakuplja



Slika 1.2: Primeri često korišćenih električnih komponenti u razvoju uređaja.

naelektrisanje. Dioda koristi simbol koji liči na strelicu koja pokazuje mogući smer kretanja struje, dok bi u suprotnom smeru struja bila zaustavljena.

Poznavanje električnih šema omogućava praćenje već dizajniranih šema koje je kreirao neko drugi, ali i praćenje načina povezivanja i rada komponenti u originalnom dizajnu.

1.4 Koraci pri dizajnu električnog kola pametnog uređaja

1. Razmisliti detaljno o nameni kola i njegovom željenom radu i osobinama.
2. Dizajnirati kolo u skladu sa izabranom primenom i napraviti blok šemu.
3. Detaljnije razraditi blok dijagram i napraviti listu potrebnih komponenti i njihovih željenih osobina.
4. Izabrati komponente u skladu sa osobinama kao što su: opseg napona, tolerancije, otpornosti, temperaturni koeficijent, efikasnost opterećenja, složenost i kompleksnost, parametri u odnosu na okruženje (vlažnost, pritisak, vibracije, temperatura okruženja), mehanički parametri (dimenzije, težina).

5. Proveriti opcije komponenti u odnosu na proizvođača: dugoročna dostupnost, podrška, dokumentacija (datasheets), softverski alati, cena, dodatne slične opcije u seriji.
6. Napraviti prototip pre proizvodnje.

1.5 Često korišćena namenska električna kola

Vrlo često nije potrebno dizajnirati kola od nule. Ovo važi kako za razvoj hardvera, tako i za razvoj softvera. Kao što postoji veliki broj dobro osmišljenih algoritama za softver, tako postoji dugogodišnji razvoj i usavršavanje namenskih kola koja je moguće koristiti prilikom razvoja pametnih uređaja.

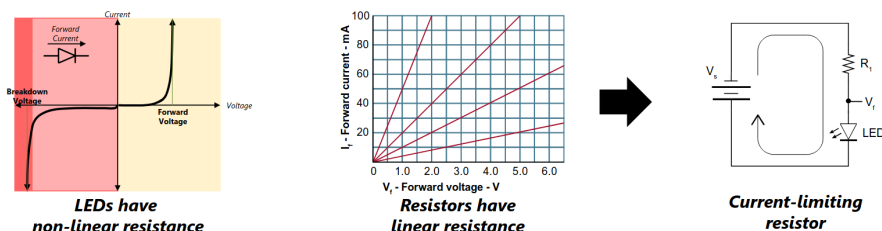
Nije neophodno memorisanje ovih kola, ali je potrebno generalno poznavanje njihovih mogućnosti i osnovnih kategorija kola. Znanje o osnovnim namenskim kolima je od značaja i prilikom razvoja originalnih rešenja. Takođe, vrlo često postoje već fabrikovana pakovanja namenskih kola u formi čipa ili štampanih ploča.

Ovde će biti izložena najčešće korišćena specijalna električna kola. Pored toga što su često korišćena, ova kola daju uvid u važne i često korišćene koncepte u uređajima.

1.5.1 Ograničavač struje otpornikom

Vrlo je važno da kolo i sve komponente u njemu budu zaštićene od prekomerne struje koja bi bila pogubna za njihov rad. Komponenta koja na jednostavan način smanjuje struju u kolu je otpornik. Na primer, u kolima sa LED diodom, zavisno od primenjenog napona na krajevima diode, dioda će pružati otpor protoku struje u jednom opsegu, međutim za napon provođenja preći će na pružanje vrlo malog otpora struji. Sa druge strane, otpornik linearno menja struju u odnosu na primenjeni napon u znatno većem rasponu napona.

Primenom otpornika u kolu moguće je sprečiti kratke spojeve ili ponašanje sličnog kratkom spoju za određene napone, kao i prilagoditi vrednost struje na željene opsege u skladu sa mogućnostima elemenata u kolu.

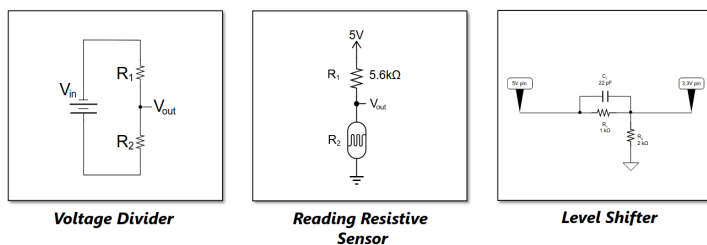


Slika 1.3: Kolo sa otpornikom i diodom za ograničavanje struje.

1.5.2 Promene napona (Delitelj napona, Regulator, Promene nivoa napona)

Često napon sa kojim se radi nije odgovarajući za komponente u kolu. Postoje kola koja pomažu da se promeni napon sa jednog nivoa na drugi.

Jedno veoma jednostavno, a efikasno kolo za promenu vrednosti napona zove se delitelj ili razdelnik napona. Ovo kolo se sastoji od dva serijski vezana otpornika. Ukupan napon obuhvata oba otpornika, ali ako se posmatra izlaz samo na jednom od njih, dobija se smanjena vrednost napona u skladu sa vrednostima otpornika i ukupnog napona na njima.



Slika 1.4: Kola za promenu nivoa napona.

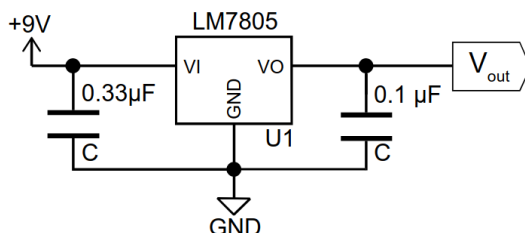
Vrednost napona i princip razdelnika napona može da se koristi i za indirektno očitavanje vrednosti sa senzora. Na primer, na mesto jednog otpornika može da se postavi fotootpornik čija otpornost zavisi od količine svetlosti. Međutim, ovu novu otpornost jednostavnije je meriti indirektno kroz očitavanje napona na fotootporniku i principa razdelnika napona.

Level shifter je kolo koje omogućava promenu nivoa napona, obično između različitih nivoa digitalnog signala. Ova vrsta kola je obično ak-

tivna.

1.5.3 Stabilizator napona

Vrlo je važno da kola imaju mali elektronski šum. Napon u kolima nije uvek savršeno ravan. Moguće je da se pojave pikovi pozitivne ili negativne vrednosti, što može ozbiljno ugroziti rad komponenti. Za poravnanje napona, odnosno za uklanjanje varijacija, kao i za eliminisanje pikova, često se koriste kondenzatori.



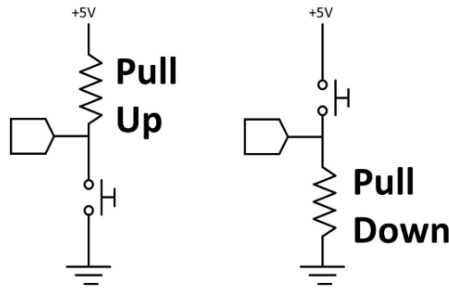
Slika 1.5: Primer stabilizatora napona.

Jedan način povezivanja kondenzatora u ove svrhe je između terminala i mase. U slučaju konstantnog DC napona, kondenzatori nemaju struju koja prolazi kroz njih (osim veoma male struje curenja). Ako dođe do velikog pika na terminalu, kondenzator će početi da se puni, a zatim će se vremenom isprazniti. Međutim, kondenzator neće dozvoliti nagle promene napona na svojim krajevima, već samo postepene. Kapacitivnost kondenzatora definiše brzinu promene napona na njegovim krajevima. Za tipične vrednosti napona na logičkim nivoima, vrednosti kapacitivnosti od 0.1µF do 10µF su dovoljne, dok bi veće vrednosti napona zahtevale i veću kapacitivnost.

1.5.4 Pull-up i pull-down otpornici (Eliminacija šuma pri otvorenoj vezi)

Prilikom povezivanja kola na ulazni signal, situaciju možemo posmatrati kao postojanje prekidača kojim je regulisana veza.

Ako je prekidač zatvoren i ulaz povezan na kolo, situacija je jasna i ulaz dolazi na kolo. Međutim, ako kolo trenutno nije povezano na ulaz i prekidač je otvoren, ostaje pitanje kolike je vrednosti napona koje stiže na kolo. Struja bi jasno bila nula jer nema mesta protoku,

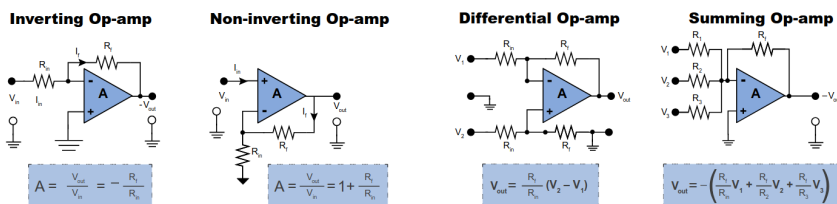


Slika 1.6: Pull-up i pull-down otpornici

međutim napon je definisan nagomilanim naelektrisanjem i zavisno od prethodnog ponašanja ili dodira rukom može doći do zadržavanja napona. Ovaj napon na otvorenom kraju čini kolo nedeterminističkim. Da bi se izbegla ovakva situacija, ovaj kraj se vezuje na masu preko pull-down otpornika. Ovo znači da kada je prekidač otvoren, ulaz u kolo je precizno jednak nuli. Ako se umesto otpornika i mase koristi otpornik spojen na pozitivni izvor napona, ovakav otpornik i napajanje takođe nude determinizam u radu, pri čemu se otpornik naziva pull-up otpornik. U ovom slučaju kada ulaz nije povezan, njegov nivo je deterministički visoke vrednosti.

1.5.5 Kola sa operacionim pojačavačem

Često postoji potreba za pojačanjem signala, bilo da bi se povećala njegova vrednost, bilo da bi se u većoj meri odvojio od šuma. Vrlo često se pojačavači analognog signala realizuju korišćenjem operacionih pojačavača. Ova kola se u šemama označavaju trouglom sa dva ulaza (invertujućim - i neinvertujućim +) i jednim izlazom koji se dobija pojačanjem razlike pozitivnog i negativnog ulaza.



Slika 1.7: Kola sa operacionim pojačavačima

Vrlo često se operacioni pojačavač ne koristi direktno, već u različitim realizacijama namenskih pojačavača. Prve dve šeme na slici predstavljaju invertujući i neinvertujući pojačavač koji pojačavaju ulazni signal uz ili bez invertovanja njegovog znaka. Vrednost pojačanja reguliše se vrednostima otpornika u kolu u skladu sa izrazima datim na slici.

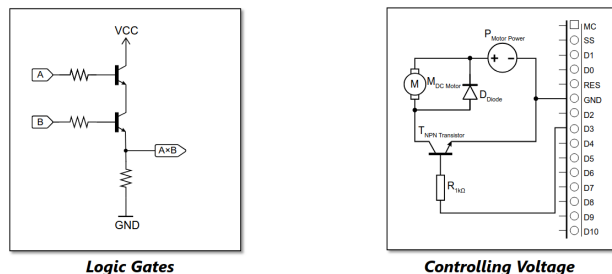
Pomoću kola sa operacionim pojačavačima moguće je i pojačavati razliku dva ulazna signala, kao u slučaju diferencijalnog pojačavača, ili kreirati sabirač ulaznih signala gde svaki ulaz ulazi u određenoj proporciji u skladu sa vrednostima otpornika, kao što to ilustruje poslednja šema na slici.

Kola sa operacionim pojačavačima koriste se za pojačanje zvuka u audioelektronici, za pojačanje izlaza raznih analognih senzora, kao i za pojačanje radio signala.

1.5.6 Tranzistorski prekidač

Za realizaciju prekidača mogu se koristiti tranzistori ili, za veće napone i struje, releji. Prekidači mogu biti kontrolisani od strane korisnika, ali se često koriste i u kolima gde se kontrolišu od strane delova kola i utiču na ukupan rad i ponašanje.

Tranzistor karakterišu tri kraja. Jedan čest način posmatranja tranzistora je u konstrukciji rada kao prekidač. U ovom slučaju se na dva kraja tranzistora nalazi napon, dok treći kraj služi da kontroliše ponašanje prekidača. Na primer, postavka može da bude da ako se na treću liniju postavi napon, prekidač se zatvori, i obrnuto.



Slika 1.8: Tranzistorski prekidač

Umesto primene kao jednog prekidača, tranzistori se mogu povezivati i u širu logiku. Na primer, sistem može da koristi dva prekidača i radi samo ako su uključena oba (I kolo). Na primer, za aktivaciju

alarma postoje dva uslova: da je alarm uključen i da je aktiviran da bi se oglasio.

Takođe se mogu koristiti za kontrolu većeg napona, kao što je prikazano na primeru sa motorom, gde ga kontroliše neka vrsta slabije struje.

Glava 2

Električni elementi pametnih uređaja

Cilj poglavlja

Cilj ovog poglavlja je detaljan pregled električnih elemenata koji se koriste u pametnim uređajima — komponentata koje koriste svetlost, elemenata za napajanje i modifikaciju napona, aktuatora za kretanje i skupa senzora za opservaciju. Naglasak je na principima rada, ključnim parametrima i primerima praktične primene.

2.1 Električni elementi pametnih uređaja koji koriste svetlost

Jedna grupa specifičnih elemenata pametnih uređaja koristi svetlost prilikom svog rada. Postoje različite namene ovakve vrste komponentata: svetlosni efekti, indikacija statusa signala i korisnički displeji.

Indikacija statusa signala može da se obavlja različitim vrstama komponenti. Ako je indikacija samo u grupi statusa gde različita boja ima različito značenje, moguće je korišćenje dioda (Light Emitting Diode). Za veću preciznost, ako je ona potrebna, moguć je prikaz u vidu brojčane vrednosti signala na ekranu. Čak i audio i video signal, odnosno njihove informacije, mogu da budu inkorporirane u okviru svetlosnog indikatora, pa i u vidu svetlosnih efekata.

Neki od parametara po kojima se ove komponente razlikuju su osvetljenost (koliko svetlosti je generisano), ugao gledanja, boja, napon

potreban za emitovanje svetlosti i struja potrebna za emitovanje svetlosti. Ovi parametri imaju svoju metriku za izražavanje vrednosti pa se tako za osvetljenje koristi kandela, ugao gledanja u stepenima, boja u formi talasne dužine tipično u nanometrima ili u formi radijacione temperature (K), napon u voltima, a struja tipično u miliamperima.

LED diode emituju svetlost, pri čemu, zavisno od njihove izrade i korišćenih materijala, postoje diode različitih boja svetlosti koju emituju. LED diode mogu da budu namenjene za emitovanje infracrvene ili za emitovanje ultraljubičaste svetlosti. Pored dioda koje emituju svetlost, postoje i fototranzistori i fotodetektor koji detektuju svetlost.



Slika 2.1: Primeri elemenata koji koriste svetlost.

LED diode mogu da budu raspoređene u funkcionalne oblike: pri radu se uključuju izabrane diode i, na primer, u slučaju sedmosegmentnog displeja kreiraju izabrani broj ili slovo. Postoje i svetlosne matrice, gde se koriste setovi LED dioda raspoređeni u obrasce za prikazivanje slika. Postoje LED diode sa željenim oblikom radi boljeg uklapanja, kao i moduli sa LED diodama za pozadinsko osvetljenje. Postoje elektroluminescentne žice za duže osvetljavanje, svetlosne cevi i LED trake. Takođe postoje različite tehnologije za proizvodnju svetlosti: sijalice sa žarnom niti, laserske diode i LCD displeji, gde je moguće prikazati tekst i grafičke sadržaje.

Svetlost se može koristiti i za signaliziranje informacija. Na primer, postoji uređaj označen kao fotoprekidač (photo interrupter). Ovaj uređaj se sastoji iz dva dela, emitira svetlosti i detektora; ukoliko se nešto nalazi između njih, dolazi do prekida u prijemu svetlosti. Na ovaj način moguće je detektovati kretanje i brzinu.

Među uređajima koji koriste svetlost pri radu postoje i optokapleri. Optokapleri su uređaji koji povezuju dva kola i omogućavaju razmenu informacija pomoću svetlosti onda kada je potrebno da kola ne budu električno spojena. Razlog za izbegavanje električnog spajanja može biti visok napon jednog kola, opasnost od vode i slično.

Postoje i uređaji koji pretvaraju različite intenzitete svetlosti u frekvencijsko ili drugi vrste kodirane signale koristeći optički kanal.

2.2 Električni elementi pametnih uređaja namenjeni modifikaciji napona

Pametnim uređajima je neophodna električna energija za njihov rad.

Za napajanje pametnih uređaja često se koriste baterije koje omogućavaju jednostavno premeštanje uređaja. Postoji mnogo različitih vrsta baterija koje se razlikuju po obliku i veličini, naponu koji obezbeđuju, trajanju rada (amper-časovima), i načinu povezivanja sa ostatkom kola. Primeri baterija i pakovanja za njihovo povezivanje mogu biti ilustrovani na datoj slici.

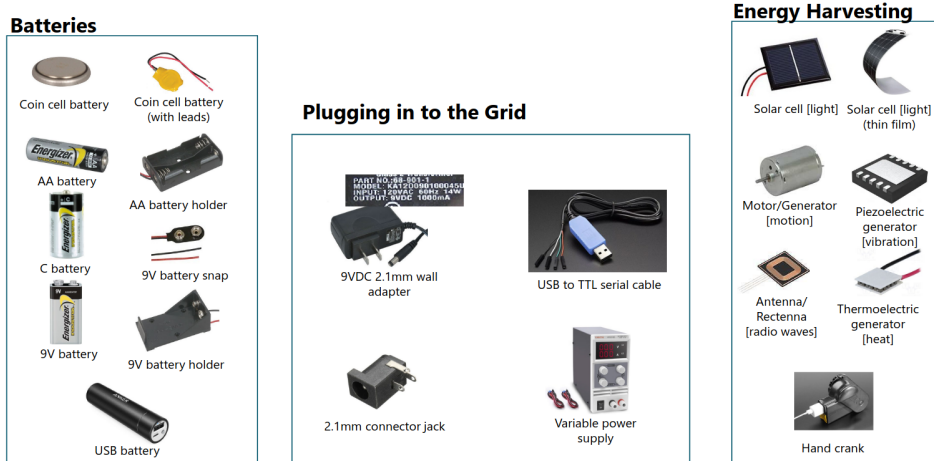
Zavisno od načina postavljanja uređaja i ako ne postoji potreba za mobilnošću, izvor energije može biti električna mreža. U tom slučaju neophodno je voditi računa o vrednosti napona koji stiže iz mreže i, ukoliko je potrebno, smanjiti ga odgovarajućim uređajima. Pored direktnog povezivanja u mreži, moguće je napajanje preko drugih izvora ili konvertora, na primer putem USB priključka.

Danas postoji veliki broj načina dobijanja energije sakupljanjem iz okoline (energy harvesting). Ovo je naročito korisno kada uređaj nije priključen na mrežu i kada se želi izbeći česta promena baterije.

Jedna grupa ovakvih izvora su solarne ćelije koje koriste svetlost i pretvaraju je u električnu energiju. Postoje ravne solarne ćelije i tanke savitljive solarne ćelije koje se mogu saviti oko predmeta.

Za generisanje električne energije mogu se koristiti i generatori koji transformišu kretanje u električnu energiju. Piezoelektrični generatori koriste vibracije u okruženju i pretvaraju ih u električnu energiju; proizvode vrlo malu struju, ali ponekad dovoljnu za određene senzore. Termoelektrični generatori transformišu toplotnu razliku u električnu energiju. Antene primaju elektromagnetne talase i transformišu ih u električni signal.

Zavisno od željenih parametara izvora električne energije, vrednost napona i kapacitet baterija mogu se modifikovati povezivanjem baterija u seriju ili paralelu. U serijskoj vezi naponi se sabiraju dok je kapacitet jednak kapacitetu jedne baterije; u paralelnoj vezi napon ostaje isti, a kapacitet se uvećava (umnožak odgovarajućih faktora).

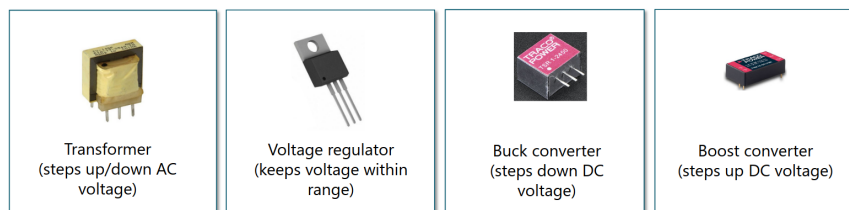


Slika 2.2: Primeri baterijskih pakovanja.

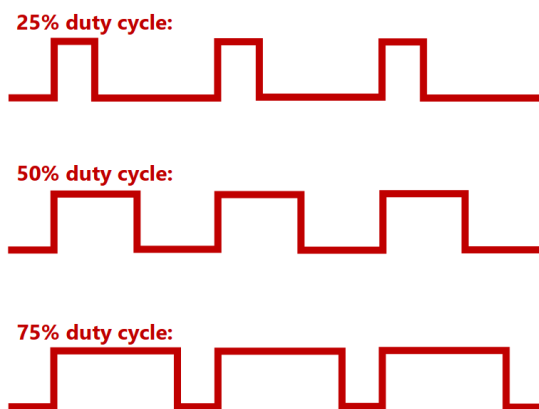
Pored mehaničkog povezivanja baterija, postoje namenski sklopovi za modifikaciju napona. Transformator može da menja naizmenični (AC) napon, podižući ili spuštajući njegovu vrednost. Regulator napona se koristi za održavanje napona u dozvoljenom opsegu, štiteći uređaj od naglih povećanja ili padova napona; regulator takođe smanjuje elektronski šum. Za smanjenje jednosmernog (DC) napona koriste se konvertori poznati kao buck konvertori, a za povećanje DC napona koriste se boost konvertori.

Prilagođavanje vrednosti izlaznog napona često se postiže i korišćenjem tehnike impulsno-širinske modulacije (Pulse-Width Modulation, PWM). Ova tehnika podrazumeva korišćenje pravougaonih impulsa koji su ili u uključenom ili u isključenom stanju, a odnos vremena provedenog u uključenom stanju u odnosu na ukupni period (duty cycle) određuje efektivnu vrednost. Ovaj način realizacije analogne vrednosti se često koristi zato što je efikasan i jednostavan. Na primer, PWM može regulisati osvetljenost LED izvora ili brzinu DC motora.

Otpornici su komponente koje pružaju otpor protoku struje i služe za zaštitu komponenti, prilagođavanje napona i merenja. Postoje termi-



Slika 2.3: Primeri komponenti koje menjaju napon.



Slika 2.4: Princip PWM signala.

stori (otpornici čija se otpornost menja sa temperaturom) i fotootpornici (otpornici čija se vrednost menja sa intenzitetom svetlosti). Potenciometri omogućavaju promenljivu vrednost otpornosti kojom korisnik može direktno da utiče na postavke uređaja.

Pored vrednosti otpornosti (izražene u omima), otpornike opisuje tolerancija (izražena u procentima), maksimalna snaga (power rating) i maksimalni napon koji mogu podneti. Boje pojaseva su jedan od uobičajenih načina označavanja vrednosti otpornika.

Kondenzatori omogućavaju skladištenje naelektrisanja i privremeno čuvanje energije. Kondenzatori ne dozvoljavaju nagle promene napona i koriste se za ravnanje napona u prisustvu pikova ili šuma. Kondenzatori imaju vrednost izraženu u faradima, toleranciju, maksimalni napon, struju curenja i ekvivalentni serijski otpor (ESR). Ove vrednosti su navedene u data-sheet dokumentaciji komponente.

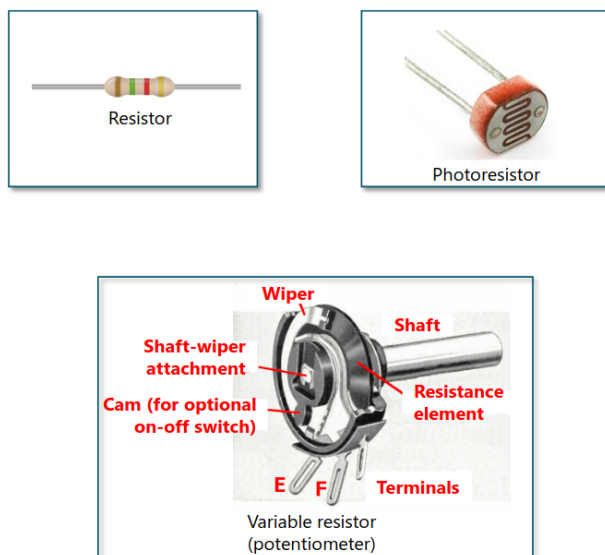
Diode dozvoljavaju kretanje struje samo u jednom smeru i imaju mnoge primene: zaštita kola (npr. flyback diode kod motora), ispravljači za pretvaranje AC u DC, i senzorska rešenja. Diode imaju nelinearnu strujno-naponsku karakteristiku, a primer parametara je napon provođenja (forward voltage), maksimalna struja provođenja i maksimalni reverzni napon (breakdown voltage). Ako se dioda izloži prevelikom reverznom naponu bez namernog dizajna za proboj, kao u slučaju Zener dioda, može doći do oštećenja.

2.3 Električni elementi pametnih uređaja za pomeranje i/ili kretanje

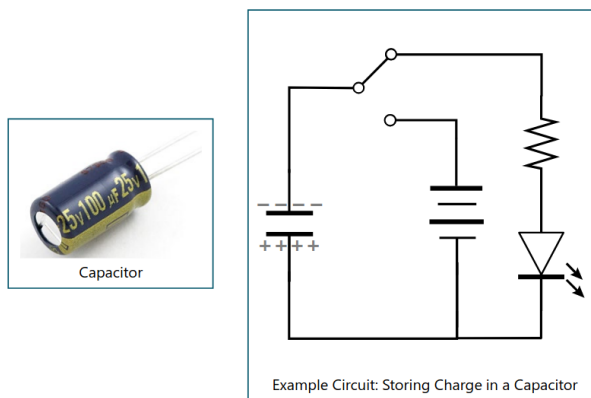
Pametni uređaji, zavisno od primene, mogu imati pomerajuće delove ili se pomerati kao celina. Ključna tehnologija za ove potrebe su motori, koji konvertuju električnu energiju u mehaničko kretanje.

Jedan često korišćen princip rada je slučaj klasičnog (brush) motora gde sistem fiksnih magneta i rotirajućeg elektromagneta čine osnovu rotacije, a promenom pravca struje polaritet elektromagneta se menja i dolazi do rotacije osovine.

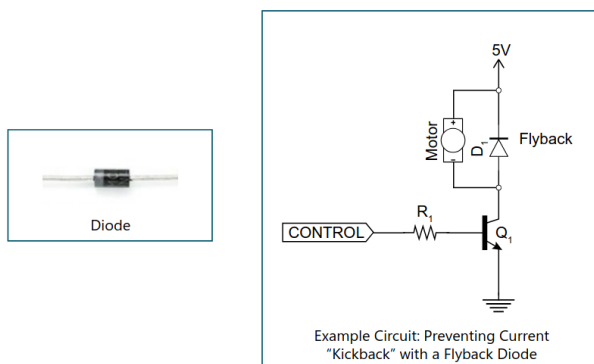
Postoje različite vrste motora, a izbor zavisi od primene. Motor sa četkicom (brushed motor) je jeftin, lako se kontroliše i daje dobar obrtni moment pri niskim RPM, ali četkice se troše i stvaraju buku. Brushless motori (bez četkica) su skuplji, pouzdaniji i efikasniji, daju veću snagu i obrtni moment, ali zahtevaju složeniju elektroniku



Slika 2.5: Otpornici – primeri različitih vrsta.



Slika 2.6: Kondenzator i primer primene.



Slika 2.7: Dioda i primer primene.

za upravljanje. Step motori (stepper) omogućavaju precizno korak-po-korak pozicioniranje i često se koriste u štampačima, CNC-ovima i robotskim aplikacijama; imaju visoku preciznost ali ograničenu brzinu i ponekad veće energetske zahteve pri zadržavanju položaja.

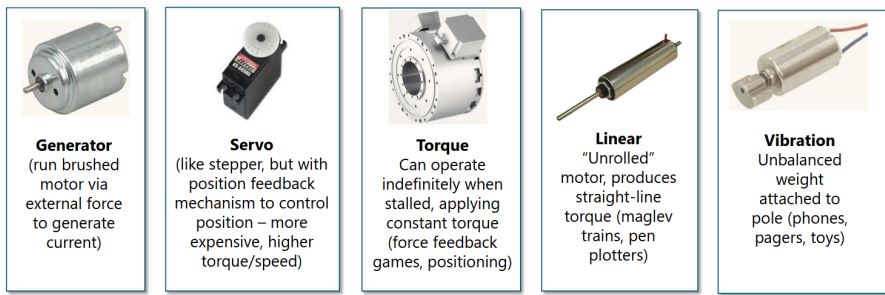
Postoje i servo motori koji kombinuju pozicioniranje sa povratnom spregom (feedback) i često pružaju veći obrtni moment pri većim brzinama u odnosu na step motore. Linearni motori omogućavaju pravolinijsko pomeranje (korisno u specifičnim industrijskim aplikacijama), dok su vibracioni motori jednostavni izvori vibracije u malim uređajima (telefoni, igračke).

Za realizaciju električkog prekidača koriste se tranzistori i releji. Tranzistori imaju tri priključka: dve linije za protok struje i jednu kontrolnu žicu. Kada se na kontrolni priključak primeni napon, tranzistor uspostavlja strujni put; kada napon prestane, prekid se uspostavlja. Tranzistori se često koriste i kao pojačavači. Releji se koriste tamo gde su potrebni prekidači za više napone i struje.

2.4 Električni elementi pametnih uređaja za opservaciju (senzori)

Senzori pametnim uređajima omogućavaju opservaciju okoline: uređaji mogu da čuju (audio), vide (slika/video) ili da mere fizičke parametre (temperatura, vlažnost, ubrzanje, orijentacija, magnetno polje, rotaciju).

Jedan primer senzora koji radi korišćenjem svetlosti je senzor boje.



Slika 2.8: Tipovi električnih motora.

Ovaj senzor detektuje i meri u vidljivom spektru; često se koristi u poljoprivredi za sortiranje plodova po boji i procenu zrelosti. Pored poljoprivrede, primenjuje se i za čitanje medicinskih traka, merenje ambijentalnog svetla i usklađivanje boja u prikazima.

Senzori koji detektuju magnetizam nazivaju se Hall senzori. Ovi senzori menjaju izlazni napon kao odgovor na prisustvo magnetnog polja. Korisni su za detekciju blizine i poravnanje sa određenom lokacijom, a mogu da detektuju i protok struje u žici (indukovano magnetno polje). Primer primene je poravnanje robota-usisivača na njegovu bazu.

Mikrofoni konvertuju zvuk u električni signal; koriste se za snimanje, detekciju audio signala, wake-word detekciju i u sistemima za prepoznavanje govora. Mogu se koristiti i za eho-lokaciju u uslovima slabe vidljivosti.

Akcelerometri mere fizičko ubrzanje objekta i koriste se za detekciju pokreta (hodanje, trčanje), detekciju pada uređaja i praćenje stanja mašina radi preventivnog održavanja. Žiroskopi mere ugaonu brzinu i orijentaciju u 3D prostoru i često se kombinuju sa akcelerometrima za bolje praćenje pokreta.

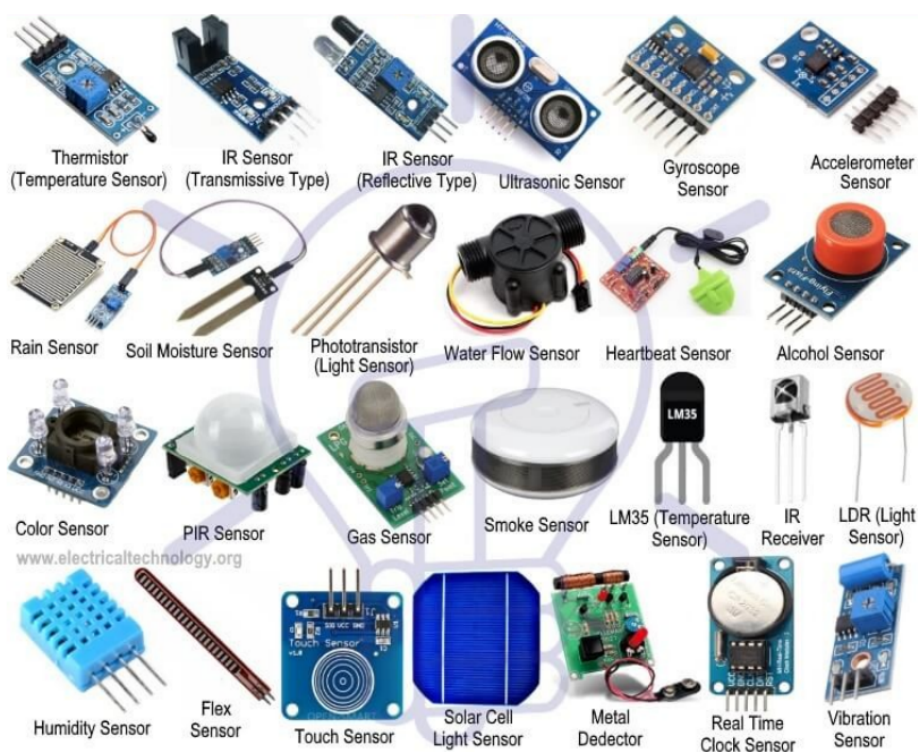
Senzori gasa detektuju sastav vazduha i mogu meriti gasove poput ugljen-monoksida, metana, benzena i drugih. Koriste se za bezbednost i za monitoring okoline. Infracrveni senzori detektuju prisustvo i kretanje u uslovima slabog osvetljenja i koriste se i u daljinskom upravljanju.

Senzori blizine (ultrazvučni, infracrveni, laserski) mere udaljenost na osnovu vremena povratka signala i koriste se za interaktivne interfejs i navigaciju robota.

U poljoprivredi, senzori vlažnosti zemljišta omogućavaju precizno navodnjavanje. Senzori dodira i kapacitivni senzori koriste se za touch

interfejsa i prepoznavanje otiska prsta. Fotootpornici mere intenzitet svetla i koriste se za adaptivno osvetljenje. Čelije opterećenja mere težinu i koriste se za detekciju prisustva ili merenje tereta. Senzori protoka mere protok tečnosti ili gasa, anemometri mere brzinu vetra, a temperaturni senzori mere temperaturu u različitim aplikacijama. Kamere snimaju slike i video i danas su ključne u primenama računarskog vida, prepoznavanju objekata i praćenju ponašanja.

Postoje i napredne i rastuće tehnologije senzora: elektronski nos (detekcija hemijskih sastava i mirisa), taktilni senzori koji mere teksturu i silu, senzori za krvne gasove (neinvazivni testovi na osnovu izdahnutog vazduha) i senzori moždanih talasa za čitanje neuroelektrične aktivnosti.



Slika 2.9: Primeri senzora.

Glava 3

Integrirana kola za pametne uređaje

Cilj poglavlja

Cilj ovog poglavlja je da objasni osnovne principe nastanka i upotrebe integriranih kola, predstavi tipične vrste i pakovanja IC-ova, opiše ključne funkcionalne kategorije često korišćenih IC-ova i da pregled modernih dodatnih (breakout / shield / HAT) ploča koje ubrzavaju razvoj ugrađenih sistema. Poglavlje obuhvata i osnovne postupke izrade integriranih kola (dopiranje, litografija) i praktične primere namenskih ploča.

3.1 Integrirana kola

3.1.1 Uvod u integrirana kola

Kreiranje kola na protoboardu za testiranje ili njihovo lemljenje radi korišćenja ima svoja ograničenja. Ovi postupci su obično spori i osetljivi, a dobijeno rešenje često zauzima prostor zavisano od veličine pojedinačnih korišćenih komponenti.

Umesto ovakvih diskretnih kola, za mnoga namenska kola koriste se njihova integrirana rešenja. U okviru integriranih kola u velikoj meri je smanjena veličina celog kola i povećana njegova osetljivost. Takođe, nakon veoma skupog originalnog dizajna jednog integriranog kola, postupak kreiranja velike količine istih kola je jeftiniji i značajno ubrzan u

odnosu na diskretna kola. Ove karakteristike dovode do činjenice da su integrirana kola praktična kada su namenjena za masovnu proizvodnju.

Kreiranje kola u integriranoj tehnologiji oslanja se na korišćenje poluprovodnika. Provodnici su materijali kroz koje struja teče lako, dok izolatori imaju izuzetno visok otpor. Ponašanje poluprovodnika može da se menja zavisno od materijala sa kojima se dopira. Prilikom izrade integriranih kola vrlo precizno se sprovodi dopiranje različitim atomima i u različitim segmentima kola i na taj način se uslovno rečeno crta kolo.

Atomi različitih hemijskih elemenata imaju različite broje protona, po čemu su i složeni u periodnom sistemu elemenata. Na primer, silicijum ima atomski broj 14, odnosno ima 14 protona. Kako broj protona odgovara broju elektrona, silicijum ima i 14 elektrona. Elektroni na najudaljenijoj orbiti se nazivaju valentni. Ovi elektroni su najslabije vezani za atom. Za silicijum važi da ima 4 valentna elektrona. Prilikom povezivanja moguće je da elektroni postanu zajednički.

Ispostavilo se da bor ima samo tri elektrona u svojoj najudaljenijoj orbiti, a fosfor ima pet. Kombinovanjem silicijuma sa borom ili fosforom moguće je dobiti kretanje elektrona kao posledica različitog broja valentnih elektrona i popunjavanja elektrona u najudaljenijoj orbiti.

Postoje dva veoma različita načina dopiranja. Jedan stvara elektronske šupljine kao kod dopiranja borom (p-tip dopiranja), a drugi stvara previše elektrona kao kod dopiranja fosforom (n-tip dopiranja). Oba načina omogućavaju da struja teče preko onoga što je ranije bio izolator.

Dopiranje materijala može da bude p ili n tipa. Međutim, moguće je i kombinovanje i spajanje ovakvih regiona čime se dobijaju nove komponente. Na primer, moguće je nacrtati diodu nadovezujući p i n region jedan na drugi ili tranzistor na primer kao n, pa p, pa opet n region uz postojanje kontrolne žice. Može da se nacrti mnogo električnih komponenti samo uzimajući poluprovodnik i dopirajući ga.

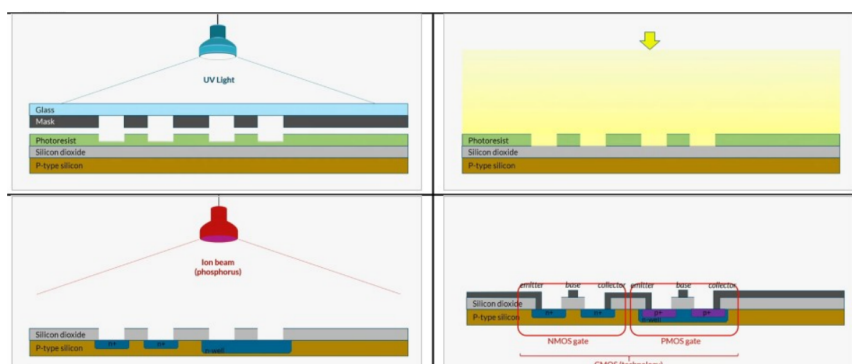
Kako se zapravo sprovodi proces dopiranja, odnosno iscertavanja dizajna? Radi se postupkom litografije gde je moguće nacrtati dizajn i smestiti ga na foto masku, a zatim ga naneti na silicijumsku pločicu. Poslednji korak predstavlja dodavanje metalnih konektora.

Na slici je ilustrovan jedan primer postupka iscertavanja na silicijumskoj pločici. U ovom primeru izabrano je dopiranje koje na kraju dobija tranzistore u CMOS tehnologiji. Za drugu vrstu kola konkretna maska i tipovi dopiranja bili bi drugačiji, ali bi idejni postupci i

način rada bili isti. U ovom primeru početna silicijumska pločica se dopira p tipom, a zatim se stavlja sloj silicijum dioksida preko njega, a onda fotorezistivni sloj preko njih. Fotorezistivni sloj je otporan na svetlost, ali može biti na neki način nagrižen svetlošću. Sledeći sloj je dizajnirana fotomaska i na kraju sloj stakla. Nakon ovoga, slojevi se izlažu ultraljubičastom svetlu što će dovesti do nagriženja fotorezistivnog sloja na određenim mestima. Sada je moguće primeniti kiselinu ili neku drugu vrstu supstance koja nagriza silicijum-dioksid na izloženim mestima. Sada je silicijum p-tipa izložen na određenim mestima, pa se na ovim otvorima može uraditi dopiranje fosforom i kreiranje n-tipa. Postupak može da se ponavlja i da se izlažu drugi delovi ili mogu da se dodaju i nova dopiranja p-tipa.

Na kraju se odozgo nanosi sloj metalnog oksida za konektovanje.

Ovaj postupak se obavlja u fabričkim pogonima integriranih kola.



Slika 3.1: Pojednostavljeni prikaz postupka izrade integriranog kola.

3.1.2 Često korišćena integrirana kola

Integrirana kola su tipično smeštena u pakovanja koja izgledaju kao mala plastična crna kućišta sa pinovima izvan njega. Postoje različite vrste pakovanja. Neka od najpoznatijih su dvostruko linijsko kućište (Dual-Inline Package, DIP).

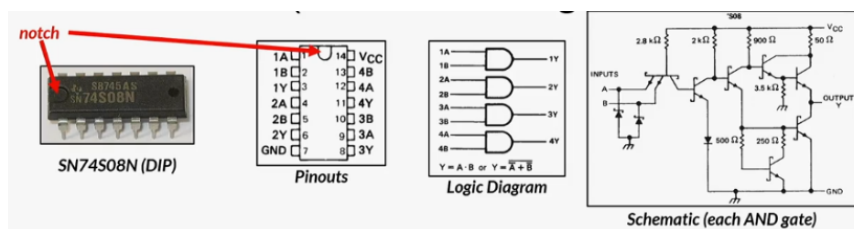
DIP pakovanje ima pinove sa obe strane kućišta koji se savijaju i spuštaju na dole. Pinovi su dizajnirani tako da se savršeno uklapaju u proto board.

U kućišta koja su dizajnirana da se postavljaju na površinu ploče (Surface-Mount Device, SMD) spadaju Small Outline IC (SOIC) i Quad Flat Package (QFP). Ova kućišta imaju kraće pinove, po dve ivice

za SOIC i po četiri za QFP. SMD kućišta su manja, ali i teža za lemljenje i povezivanje.

Postoji ogromno tržište za integrirana kola, kao i mnogo različitih integriranih kola.

Kao primer jednog integriranog kola biće predstavljeno četvorostruko AND kolo napravljeno od strane Texas Instruments-a. Na slici je prikazan deo podataka koji je tipično dostupan za IC. Kompletne podatke se nalaze u datasheet-u, koji se uobičajeno lako nalaze na osnovu oznake kola. Prvi deo slike daje izgled kućišta, ovde DIP. Kućište ima na neki način postavljen zarez ili tačku koja pomaže u numeraciji pina. Prvi pin je levo od zareza, a ostatak se dobija u smeru suprotnom od smera kazaljke na satu. Brojevi pinova su prikazani na drugoj slici. Logički dijagram desno od toga pokazuje kako su pinovi povezani u smislu interne logike. Na primer, pinovi 1A i 1B predstavljaju ulaze prvog logičkog AND kola, a 1Y predstavlja njegov izlaz. Na poslednjem delu slike data je i električna šema jednog od AND kola.



Slika 3.2: 7408 integrirano kolo.

U nastavku će biti dati primeri često korišćenih integriranih kola.

Jedno od najpopularnijih integriranih kola na celom svetu je 555 tajmer IC. Ovo kolo se vrlo često koristi kada je potrebno obezbediti događaje u određenom ravnomernom trajanju (astabilna konfiguracija) ili događaj posle određenog vremena (monostabilna konfiguracija). Postoji mnogo varijanti ovog IC, kao što su XR2206, I8038 i slično. Oni se razlikuju u pogledu specifičnih konfiguracija koje je moguće napraviti.

Još jedno izuzetno popularno integrirano kolo je operacioni pojačavač 741. Postoje i alternative koje se razlikuju u pogledu koji napon koriste i slično, na primer LM324.

Postoje i integrirana kola za regulatore napona. Jedan veoma popularan je 7805 čiji je izlazni napon regulisan na 5 V.

Postoji veliki broj IC za logička kola. Neki od primera su 7408 za AND kola, 7404 za NOT kola, 7486 za XOR i dr.

Još jedan popularan tip integrisanog kola se naziva registar pomeranja. Ovo kolo može imati jedan ulaz i niz izlaza i može taktovati skup bitova. Ovo je veoma korisno ako postoji potreba da se istovremeno kontroliše na određeni način niz izlaza, pomoću jednog ulaza. Na primer, može da se koristi u situacijama kada postoji namenski računarski segment sa svojim izlazima, ali još veći broj elemenata koje je potrebno kontrolisati. Registar pomeranja skladišti setove bita prilikom svog rada.

A/D konvertor (ADC, Analog-to-Digital Converter) je kolo koje konvertuje analogni ulaz u digitalni izlaz. Tako u slučaju pametnih uređaja, kada se očitavaju fizičke veličine one su u obliku analognih signala. Ponekad sam senzor sadrži ADC i radi konverziju, a ponekad je neophodno koristiti ADC između analognog senzora i namenskog upravljačkog sistema, kao što je mikroračunar. ADC pretvara kontinualnu vrednost analognog signala u digitalni zapis diskretnog skupa mogućih vrednosti u skladu sa svojom rezolucijom.

Postoje i D/A konvertori (DAC, Digital-to-Analog Converter) koji konvertuju digitalnu vrednost u analogni izlaz. Na primer, u slučaju kontrole rada motora postoji potreba da motor menja napon i svoju brzinu kontinualno.

Komparatori su integrirana kola koja imaju dva ulaza koja porede, a kao svoj izlaz daju binarnu digitalnu vrednost u skladu sa znakom razlike. Mogu se koristiti i za generisanje pravougaoone povorke impulsa.

Još jedna korisna vrsta integrisanog kola je drajver motora. Drajveri motora rade tako što prihvataju ulaz koji može biti analogni signal ili digitalni prikaz onoga što je potrebno da radi, a zatim mogu da pokreću motor. Neki drajveri motora su veoma moćni, mogu da pokreću veoma velika induktivna opterećenja, ogromne elektromagnete i stepere motore.

Postoje i LED drajveri koji su slični pomeračkim registrima. LED drajveri omogućavaju istovremenu kontrolu jednim ulazom velikog broja LED dioda i omogućavaju im da budu na različitim nivoima osvetljenosti i intenziteta, da budu uključene ili isključene i slično.

Postoje i I/O ekspanderi, što su druge varijante pomeračkih registra koji taktuju veliki broj bitova, i kontrolišu veliki broj linija koristeći jedan ulaz.

Wi-Fi moduli se koriste kada je potrebno da uređaj komunicira

bežično sa drugim uređajima koristeći IEEE 802.11 standard. Ovo je veoma kompleksno kolo po svom sadržaju, ali danas može da se kupi kao IC.

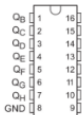
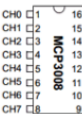
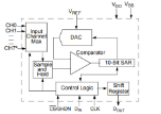
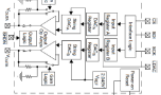
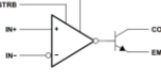
Ovo su neki primeri integriranih kola, a ima ih još mnogo. Dobro je pre dizajna novog originalnog, posebno složenog kola, proveriti da li već postoji direktno takvo ili blisko IC kolo. Vrlo često IC kola, zbog proizvodnje u velikim količinama, imaju i nisku cenu i relativno su jednostavna za korišćenje. Međutim ako se radi nešto što je zaista novo i nepoznato što se ranije nije pojavljivalo, biće teško pronaći integrirana kola za to. Treba voditi računa da su IC obično dizajnirana sa malim nivoima napona, odnosno da je u slučaju rada sa višim vrednostima napona i struja potrebno tražiti druga rešenja.

IC family	Type	Pinout	Logic diagram
555	Timer		
741	Operational Amplifier (Op-Amp)		
78xx	Voltage Regulator		
74xx	Logic Circuits		

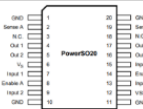
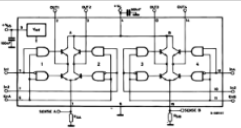
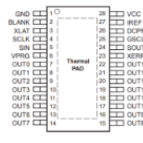
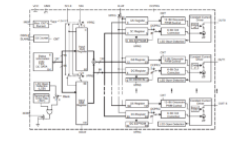
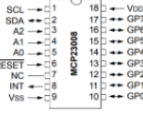
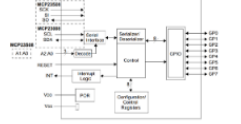
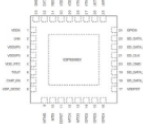
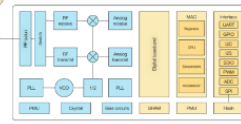
Slika 3.3: Primeri integriranih kola – tajmer 555, operacioni pojačavač 741, regulator napona 7805 i logičko AND kolo 7408.

3.2 Dodatne ploče namenskih uređaja

Pored integriranih kola, danas postoji još jedan način za razdvajanje dizajna na komponente. Poslednjih desetak godina razvijen je

IC family	Type	Pinout	Logic diagram
74HC595	Shift Register		
MCP3008	A/D Converter		
MCP4822	D/A Converter		
LM311	Comparator		

Slika 3.4: Primeri integriranih kola – registar pomeranja, A/D konvertor, D/A konvertor i komparator.

IC family	Type	Pinout	Logic diagram
L298D	Motor Driver		
TLC5940	LED Driver		
MCP23008	I/O Expander		
ESP8266	WiFi module		

Slika 3.5: Primeri integriranih kola – drajver motora, LED drajver, I/O ekspander i Wi-Fi modul.

veliki broj ploča za obavljanje namenskih složenijih funkcija (breakout boards). Ove ploče se koriste kao jedna celina, jedna komponenta u većem uređaju, iako je njihova namena i sadržaj često kompleksniji. Ove ploče mogu da uključuju i IC.

Breakout ploče su obično na PCB (Printed Circuit Board) sa setom komponenti i setom pinova koji se koristi za povezivanje sa ostatkom dizajna.

Ploče sa namenskom funkcijom ubrzavaju postupak dizajna i nude pouzdano obično dobro dokumentovano rešenje. Ova rešenja često nemaju visoku cenu i zauzimaju manje prostora nego što bi se postiglo pojedinačnim rešenjem. Mana ovih rešenja je što su često namenski okrenuta za uklapanje sa jednim specifičnim procesorom, na primer mikroracunarom kao što je Raspberry Pi ili mikrokontrolerom kao što je Arduino. Ovo dalje predstavlja ograničenje i u protokolu i načinu komunikacije koji koriste prilikom razmene podataka sa glavnom procesorskom pločom. Ovo znači da namenski sistem će ponekad raditi i za drugi procesor. Takođe, namenske ploče mogu da imaju dodatne zahteve, na primer za napajanje. Rešenje koje se dobija na ovaj način

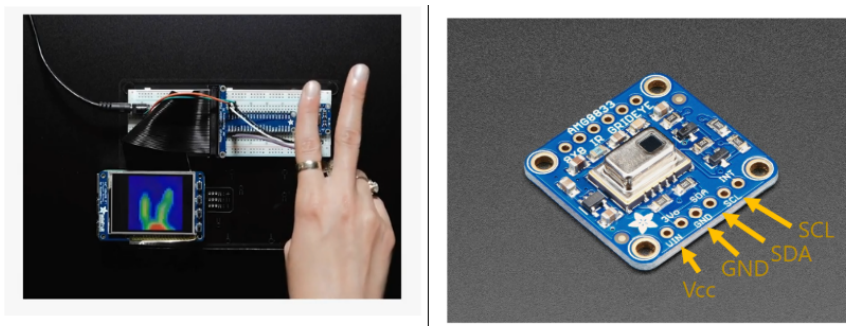
može da bude odgovarajuće i veoma brzo, međutim ograničava dizajn na tačno ono što je ponuđeno.

Postoje i namenske ploče razvijene direktno kao dodatak na postojeći mikrokontroler ili mikroračunar. Ovakve ploče se znatno teže uklapaju u drugu vrstu uređaja u odnosu na onu za koju su namenjeni, dok se naravno odlično uklapaju u one za koju su namenjeni. Ove vrste namenskih ploča imaju drugačiju opštu oznaku. Umesto breakout boards, označavaju se ponekad kao shields (na primer za Arduino ploče) ili kao HAT — Hardware Attached on Top (za Raspberry Pi ploče).



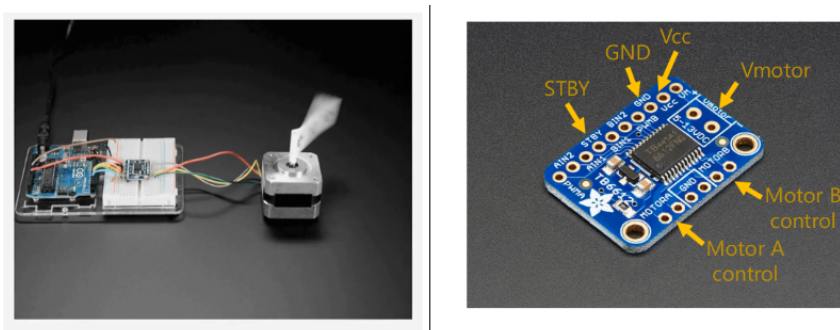
Slika 3.6: Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout (BNO055) — primer načina korišćenja i izgled ploče.

Primer jedne namenske dodatne ploče je Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout BNO055. Ova ploča ima mogućnost da meri pomeranje u 3D prostoru. Ploča za obavljanje ove funkcije koristi kombinaciju žiroskopa i akcelerometra.



Slika 3.7: Adafruit AMG8833 IR Thermal Camera Breakout Board — primer načina korišćenja i izgled ploče.

Sledeći primer namenske ploče je ploča sa IR (infrared) kamerom. Ova vrsta kamere ima velike mogućnosti primene, kao što su detektovanje kretanja u mraku, ali i detektovanje toplotnih obrazaca na ljudskom licu u većoj meri nego što to može obična kamera, pa time i mogućnost za analizu stanja i detekciju prisustva. Primer jedne ovakve namenske ploče je Adafruit AMG8833 IR Thermal Camera Breakout Board. Ova ploča sadrži Panasonic AMG8833 termalni IR senzor u 8×8 nizu koji može da detektuje ljude na udaljenosti do oko 7 m sa 10 Hz frame rate. Pored senzora, ploča sadrži i 3.3 V regulator i potrebne otpornike i kondenzatore. Ploča sadrži pinove za povezivanje na napajanje (Vin i GND) i komunikaciju sa drugim pločama (serijski takt SCL i serijski podaci SDA). Dodatno, ploča ima mogućnost uvođenja prekida (interrupt označen sa INT).

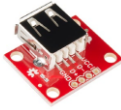
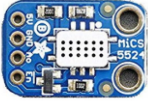


Slika 3.8: Adafruit TB6612 1.2A DC/Stepper Motor Driver Breakout Board — primer načina korišćenja i izgled ploče.

Postoje i ploče za kontrolu step motora. Može se koristiti na primer za vođenje motora u kružnom kretanju, vođenju unazad, naglom zaustavljanju. Ovo se dalje može koristiti za upravljanje robotom, kao i za skeniranje. Primer jedne ovakve ploče je Adafruit TB6612 1.2A DC/Stepper Motor Driver Breakout Board. Ova ploča sadrži integrisano kolo Toshiba TB6612 motor driver, ali i dva H-mosta, zaštitu polariteta FET, pull-up otpornike, kick-back diode. Ovaj drajver step motora može da vodi jedan ili dva motora sa 1.2 A sa 3 A vršnom vrednošću na 20 ms, upravlja smerom i brzinom, zaustavljanjem/kretanjem. U kolu je izolovan napon motora od logičkog napona. Neki od pinova motora označeni su na slici.

Sledeća tabela daje interesantan pregled kategorizacije namenskih ploča. Jedna podela bi bila: aktuatori, ekrani, konektori ili komunika-

cijski moduli, senzori.

Type	Name	Picture
Actuator	Allegro A4988 Stepper Motor Driver	
Display	SparkFun Color LCD Breakout Board	
Connector	SparkFun USB Type A Female Breakout	
Sensor	Adafruit MiCS5524 CO, Alcohol and VOC Gas Sensor Breakout	

Slika 3.9: Kategorizacija dodatnih namenskih ploča prema funkciji.

Pri korišćenju dodatnih ploča važno je obratiti pažnju na kompatibilnost napajanja, logičkih nivoa (3.3 V vs 5 V), komunikacijske protokole (I²C, SPI, UART), i softversku podršku. Gotove ploče znatno ubrzavaju prototipizaciju, ali dizajner mora voditi računa o ograničenjima koja proizlaze iz njihove specifične konfiguracije i podrške.

Integrisana kola i dodatne ploče su temelj savremenog razvoja pametnih i ugrađenih sistema. Razumevanje osnovnih principa izrade IC-ova, pakovanja, tipičnih funkcionalnosti i praktičnih namenskih ploča omogućava brži i sigurniji prelazak od prototipa do serijske proizvodnje. Pri tome je korisno prvo proveriti postojeće komponente i ploče pre ulaganja u sopstveni dizajn, voditi računa o električkim interfejsima i specifikacijama, i koristiti datasheet dokumentaciju za pravilnu integraciju.

Glava 4

Procesori i mikrokontroleri

Cilj poglavlja

Cilj poglavlja je prikaz uloge programabilnih integriranih kola (mikrokontrolera), njihovih osnovnih komponenti i principa rada, kriterijuma izbora i primere često korišćenih arhitektura, uz ilustracije i oznake.

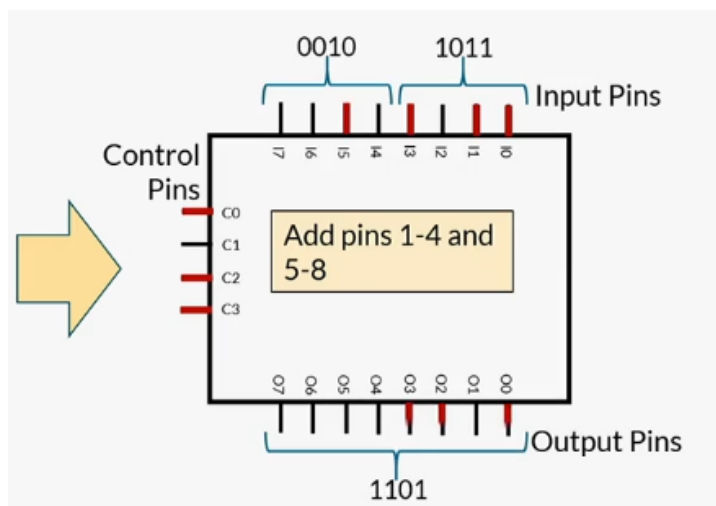
4.1 Uvod

Integrirana kola mogu da urade veliki broj specifičnih poslova za koje su namenjena. Na primer, mogu da se koriste za regulaciju napona, za povećanje vrednosti napona između dve željene izabrane vrednosti, vode rad motora i drugo. Međutim, pametni uređaji mogu da imaju i složeniju namenu, kao što su prepoznavanje lica, uzimanje i prenos objekta na željeni način, displeji za prikazivanje različitih informacija, upravljanje radom drona i prepoznavanje objekata tokom leta, prepoznavanje celina, na primer silueta ljudi prilikom vožnje. Poveravanje ovakvog posla jednom integrisanom kolu specijalne namene ne bi bilo odgovarajuće.

Međutim, šta kada bi postojalo integrirano kolo koje ima ulazne, izlazne i kontrolne pinove koji mogu da obavljaju različitu funkciju zavisno od toga kako je kolo programirano? Odnosno, kolo bi jedno vreme imalo jednu funkciju, a nekada kasnije bi moglo da ima potpuno drugu funkciju.

Jedan primer ideje programabilnog integrisanog kola je dat na Slici.

U ovom slučaju kolo se ponaša kao sabirač, odnosno kontrolnim bitima se kontroliše da se vrednosti ulaznih bita sabiraju i daju na izlazu.



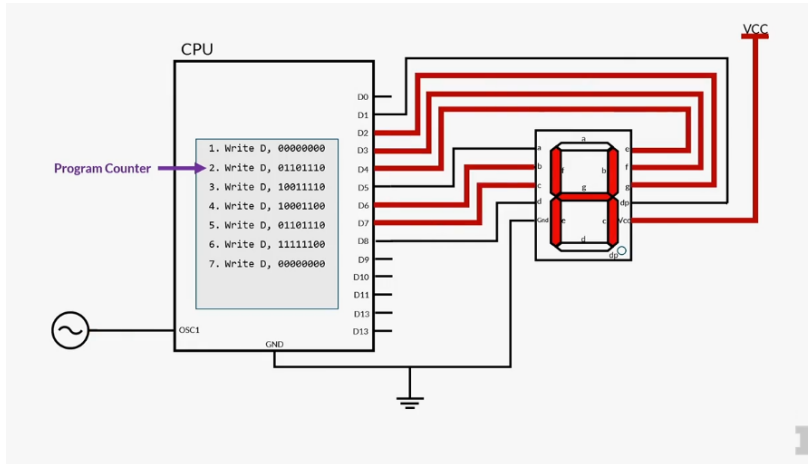
Slika 4.1: Primer programabilnog integrisanog kola.

Umesto datog primera sabiranja, reprogramiranjem kolo može preći na obavljanje drugih funkcija. Na primer, kolo može vrednost određenog pina da podiže i spušta određen broj puta ili kolo može da postane pojačavač i da vrednost jednog pina poveća određen broj puta i pošalje na drugi pin.

Integrisano kolo koje može da se programira i koje je veoma fleksibilno označava se sa CPU (Central Processing Unit). Jedan način korišćenja CPU je kada su je povezana na druga elektronska kola i koristi se za njihovo upravljanje.

Dat je primer povezivanja CPU sa drugim elektronskim sistemom, ovde sedmosegmentnim displejom, koji će dalje upravljati radom displeja. Upravljanje može da bude u etapama, odnosno da postoji više instrukcija u nizu. Kretanje kroz instrukcije prati programski brojač. Pomeranje brojača sa jedne instrukcije na drugu je podpomognuto oscilatorom koji diktira tempo, odnosno brzinu rada. Na ovaj način, jedna instrukcija može da ispiše jedno slovo na displeju, sledeća instrukcija sledeće itd.

CPU sadrži set komponenti unutar same jedinice, a za rad su mu često potrebne i dodatne spoljašnje komponente kao što su memorije i ulazno/izlazni uređaji. Tipično je memorija podeljena na memoriju za instrukcije i memoriju za podatke. Komunikacija između CPU

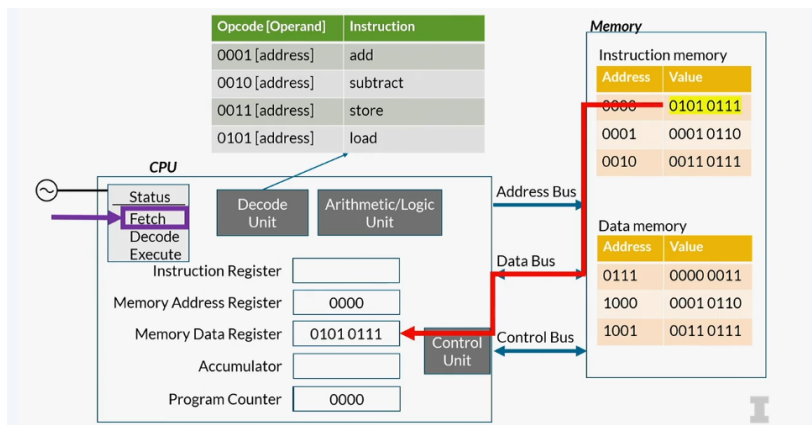


Slika 4.2: Primer povezivanja CPU sa sedmosegmentnim displejom i osnovni tok instrukcija.

i memorije obavlja se pomoću seta magistrala. Na slici su označene kao Address Bus, Data Bus i Control Bus. Preko adresne magistrale šalje se adresa u memoriji koja se posmatra. Magistralom za podatke šalju se i primaju podaci iz memorije. Kontrolnom magistralom se šalje informacija memoriji šta se trenutno obavlja.

Unutar CPU takođe postoji memorija. Postoje registri i brojači unutar CPU koji privremeno skladište vrednosti. Takođe, CPU sadrži i podatke o stanju statusa operacije koju trenutno izvršava. U primeru datom na slici prvi status je **fetch** za preuzimanje podatka i instrukcije, zatim **decode** za dekodovanje instrukcije i na kraju **execute** za izvršenje. CPU sadrži odvojene jedinice za dekodovanje (Decode Unit) koja služi za razumevanje instrukcija, jedinicu za aritmetičko-logičke operacije (Arithmetic Logic Unit, ALU) za izvršavanje operacija i kontrolnu jedinicu (Control Unit) koja kontroliše eksterne uređaje kao što je memorija.

Primer početka rada je prikazan na slici. Prva operacija je **fetch** i vrednost brojača je 0000. Ova vrednost se zatim smešta u registar za memorijske adrese, a onda i šalje preko magistrale za adrese, pri čemu je na kontrolnoj magistrali označena operacija čitanja. Vrednost u memoriji koja se nalazi na adresi je 01010101, pri čemu se ova adresa nalazi u memoriji za instrukcije. Iz ovog razloga kada podaci stignu u registar za podatke iz memorije, biće onda smešteni i u registar za instrukcije. Ovo je prikazano na Slici.



Slika 4.3: Ilustracija operacija **fetch** — preuzimanje instrukcije iz memorije, transfer preko magistrala i skladištenje u registar instrukcija.

Ako se za opcode i dekodovanje instrukcije koriste prva četiri bita otkriva se instrukcija **load**. Ovo znači da je potrebno učitavanje podataka. Kasnije se ovi učitani podaci mogu koristiti kao operand u drugim operacijama, na primer za sabiranje. Sada se učitava vrednost **0111**, odnosno smešta u registar za adrese u memoriji.

Sada se prelazi na treći status, odnosno na **execute**. Na adresi **0111** se čita **00000011** i vraća na CPU.

Ova vrednost se smešta u registar za memorijske podatke. Kako ovaj put nije u pitanju podatak iz memorije za instrukcije, već podatak iz memorije za podatke, CPU neće vršiti dodatno smeštanje u registar za instrukcije, već u akumulator za dalje operacije.

Broj opkodova, instrukcija, komponenti unutar CPU, magistrala i njihovih karakteristika, veličine adresa su neki od parametara po kojima se razlikuju različiti CPU.

Zahvaljujući procesorima moguće je mnogo toga, od prilagođenih operacija, operativnih sistema do prepoznavanja lica u mašinskom učenju. Generalno procesori omogućavaju veliki broj složenijih operacija.

Pored velike primenljivosti i mogućnosti, procesori imaju i svoja ograničenja i otežavajuće faktore. Jedna od mana je njihova veća cena u odnosu na druga elektronska kola. Naravno, postoji raspon mogućnosti, ali generalno su dizajnirani da rade opšte poslove i da se prilagođavaju različitim aplikacijama. Razvijanje i mogućnosti rada složenijih operacija i aplikacija vodi i do veće potrošnje energije. Smanjenje mogućnosti, smanjenje brzine rada i složenosti aplikacija koje

procesor može da sprovede oboriće i parametre ovih otežavajućih karakteristika, odnosno cene i potrošnje. Postoje procesori posebne namene koji su dizajnirani za specifične aplikacije koje su jeftinije i to su oni koji se nazivaju mikrokontroleri.

Dakle, mikrokontrolere karakteriše manja potrošnja, manja brzina rada, odnosno manji takt, manje mogućnosti prenosa, jednostavniji dizajn, manje unutrašnjih komponenti, manji broj instrukcija, kao i usmerenost na određene specifične aplikacije.

Mikrokontroleri tipično imaju konsolidovan dizajn gde su memorija, oscilatori i I/O kontroleri i drugi segmenti na samom čipu procesora. Ovo čini arhitekturu mikrokontrolera jednostavnijom i jeftinijom. Kako mikrokontroleri obično imaju namenske funkcije, često prateće funkcije u skladu sa posebnom namenom budu ugrađene u sam čip. Primeri ovakvih namenskih funkcija su A/D i D/A konvertori, kola za PWM (Pulse-Width Modulation), NCO (Numerically Controlled Oscillator) ili programabilne logičke blokove.

4.2 Kako izabrati mikrokontroler?

Postoji više različitih faktora koje je potrebno uzeti u obzir prilikom izbora mikrokontrolera. Jedan od glavnih je samo jezgro mikrokontrolera. Ovo je obično najsloženiji deo i definiše i mogućnosti u pogledu logike, dekodovanja, instrukcija, ALU i slično. Dizajn jezgra je obično u vlasništvu određene kompanije, bilo direktno one koja i proizvodi mikrokontroler bilo druge kompanije. Jedan veoma često korišćen intelektualno vlasnički dizajn je ARM. Njihova 32-bitna jezgra mikrokontrolera se proizvode od strane velikog broja proizvođača mikrokontrolera. Drugi čest primer jezgra je Intel 8051.

Sledeći parametar koji je potrebno razmotriti je da li je potreban mikrokontroler velikih performansi (obično sa višom cenom i višom potrošnjom) ili mikrokontroler sa veoma ograničenim resursima. Ovo je teško pitanje, na koje ponekad na početku izrade projekta nema potpuno preciznog odgovora. Zato, često proizvođači u ponudi umesto jednog rešenja nude celu seriju sa rasponom parametara. Izbor rešenja sa bogatom serijom obično vodi i ka višoj ceni (jer je skuplje i napraviti veću seriju), međutim, omogućava veću slobodu prilikom prelaska sa jednog na drugo rešenje unutar serije. Kretanje u seriji je naravno poželjno, jer je rad sa njima veoma sličan, pa je put prilagođavanja sa

jednog na drugo rešenje značajno skraćen.

Pakovanje mikrokontrolera i njegova veličina su sledeći faktor za izbor rešenja. Neki od primera pakovanja su wafer level CSP (Chip Scale Packaging) 2 mm×2 mm pakovanje kao jedno od najmanjih pakovanja i DIP (Dual Inline Package) kao jedno od većih pakovanja.

Različiti mikrokontroleri imaju različite sposobnosti komuniciranja i odgovarajućih periferija. Neki od njih imaju unutrašnje oscilatore, dok će drugi zahtevati oscilator povezan interfejsom. Neki imaju višestruke kanale za prekide, neki ne. Neki od njih mogu da imaju direktan pristup memoriji (Direct Memory Access, DMA). Neki mogu da imaju različite načine za kontrolu konfiguracije napajanja.

Još jedna stvar o kojoj treba razmišljati je softversko i hardversko razvojno iskustvo. Ovo se ogleda u alatima za otklanjanje grešaka, emulatorima, potencijalnom postojanju razvojnih platformi, ali i u njihovim cenama.

U pogledu performansi, mikrokontroler ima različite metrike, kao što su korišćenje energije, operacije tokom jednog ciklusa takta, veličina koda, broj instrukcija i drugo.

4.3 Najpoznatiji mikrokontroleri

Postoje određeni mikrokontroleri koji su popularniji od drugih. Jedan od najpoznatijih proizvođača mikrokontrolera je Atmel, koji obezbeđuje AVR platformu. Atmel AVR dizajn karakteriše mali broj opcode-ova. Prednost ovakvih redukovanih instrukcija je fokus na brzom izvršavanju. Ova platforma podržava veliki broj single-cycle operacija. Množenje traje dva ciklusa, što je značajno manje nego za mnoge druge platforme. Druga prednost ove platforme je veliki broj registara, što je dobro za kompajlere. Mane Atmel AVR platforme su postojanje samo jednog prioriteta za prekide i mala brzina takta u odnosu na slične mikrokontrolere. Neki od njihovih najpoznatijih mikrokontrolera su ATtiny85, koji se koristi za Arduino, ATtiny1616, ATmega328.

Drugi veoma poznati proizvođač mikrokontrolera je Microchip sa PIC platformom. Microchip PIC takođe koristi RISC dizajn, odnosno redukovani set instrukcija i ima brzo izvršavanje instrukcija. Prednost ove platforme je veliki broj periferija. Mane PIC dizajna su postojanje samo jednog registra i nepostojanje čipa za debugging. Primeri njihovih mikrokontrolera su PIC10, PIC12, PIC16, PIC18, PIC24 i PIC32.

ARM proizvodi IP core dizajn za koji daje licence i prodaje drugim proizvođačima čipova. ARM se prvo bavio proizvodnjom CPU visokih performansi, a onda je uključio mikrokontrolere u okviru svoje M-series platforme (npr. Cortex-M). Imaju više vektora za prekide i više prioriteta za prekide. Karakteriše ih i veći broj registara (npr. 13 ili više). Implementacija koda obično zauzima dosta prostora, a postoji i primetnije vremensko kašnjenje prilikom implementacije.

Intel 8051 je veoma popularna platforma. Za razliku od prethodnih, ovde se koristi CISC instrukcijski skup, odnosno postoji dosta većih instrukcija. Intel 8051 ima veliki broj registara, 64 KB program/RAM adresabilnosti. Platforma je poznata po veoma brzim prekidima, što je čini dobrim izborom za sisteme u realnom vremenu. Mana platforme je korišćenje 8 bita, u odnosu na današnje jeftine 32-bitne mikrokontrolere.

Mikrokontroleri predstavljaju kompromis između fleksibilnosti i efikasnosti: nude dovoljno programabilnosti za širok spektar aplikacija uz manju potrošnju i nižu cenu u odnosu na opšte-namenske CPU jedinice. Ispravan izbor mikrokontrolera zahteva balans između zahteva aplikacije (performanse, periferije, potrošnja) i dostupnosti alata i podrške.

Glava 5

Programabilna kola

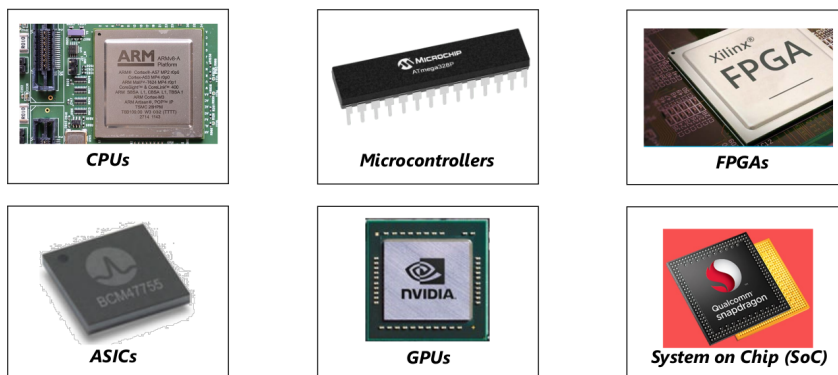
Cilj poglavlja

Cilj poglavlja je da predstavi vrste programabilnih kola koja se koriste u računarskom segmentu pametnih uređaja, objasni njihove osnovne karakteristike, prednosti i ograničenja i ilustruje tipične primene i povezivanje u savremenim sistemima. Poseban akcenat stavljen je na FPGA, ASIC, GPU, SoC i specijalne procesorske jedinice za akceleraciju AI/ML i video.

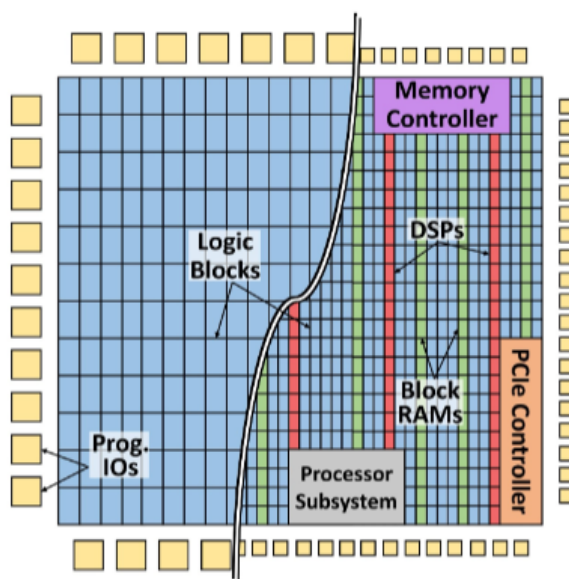
Računarski segment pametnih uređaja pored CPU i mikrokontrolera može da sadrži i druge vrste programabilnih kola. Pored mikroprocesora i mikrokontrolera tu su i FPGAs (Field-Programmable Gate Arrays), ASICs (Application-Specific Integrated Circuits), GPUs (Graphics Processing Units) i SoC (System on Chip). Navedena računarska kola imaju različite karakteristike u pogledu svoje fleksibilnosti i efikasnosti.

5.1 FPGA

FPGA omogućavaju namensku konfiguraciju kola, u skladu sa željenom primenom, pri čemu se brzina rada i struktura prilagođavaju izabranoj primeni. Osnovni elementi moderne arhitekture FPGA kola su heterogeni: pored programabilnih nizova ili programabilnih blokova (kao što su osnovna logička kola poput AND, OR, mnoge složenije kombinatorne veze, Look Up Tables - LUT i flip-flopovi), međusobnih veza blokova koje se uključuju ili ne, sadrže se i specifični namenski blokovi (na primer blokovska RAM ili DSP blokovi).

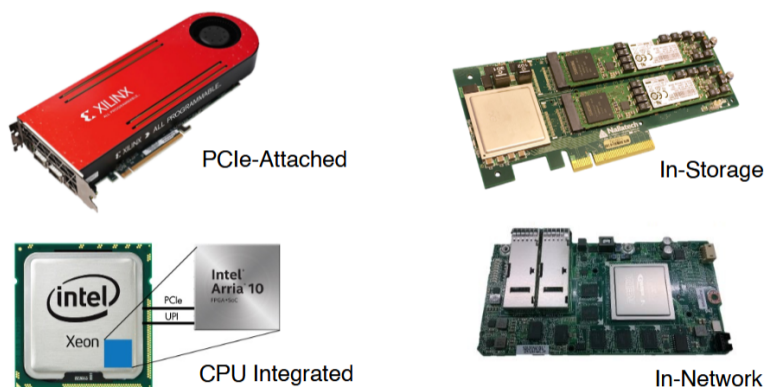


Slika 5.1: Pregled programabilnih kola.



Slika 5.2: Šematski prikaz tipične FPGA arhitekture.

Pakovanje ovakvih FPGA kola u SoC često uključuje i dodatne komponente, što je ilustrovano na slici ispod.



Slika 5.3: Primer integracije FPGA elementa unutar SoC okruženja.

Glavne prednosti korišćenja FPGA kola su velika paralelnost prilikom rada (za razliku od klasičnih CPU i mikrokontrolera), mogućnost rekonfiguracije već programiranog kola (na primer za unapređenje ili ispravljanje uočenih grešaka), mogućnost prilagođavanja u skladu sa željenom tačnošću i brzinom rada, malo i predvidljivo kašnjenje i veća energetska efikasnost u odnosu na opšte-namenske procesore za određene zadatke.

FPGAs su posebna vrsta programabilnih logičkih kola. Prilikom kupovine FPGA kola dobija se „prazan“ hardver spreman za namensko prilagođavanje. Samo prilagođavanje može da se obavlja na različite načine; među najpoznatije spadaju programiranje namenskim jezicima za hardverski dizajn (HDL — Hardware Description Language) kao što su Verilog i VHDL i dizajn korišćenjem namenskih CAD alata. Mana FPGA kola je često visoka cena — uobičajeno mogu koštati stotine ili više stotina evra, zavisno od klase i kapaciteta. Glavni proizvođači za FPGA kola su Xilinx (sada deo jedne grupe), Intel (ranije Altera) i drugi.

5.2 ASICs

ASIC su integrisana kola kreirana za određenu specifičnu aplikaciju. Ako je scenario aplikacije specifičan i postoji budžet za kreiranje

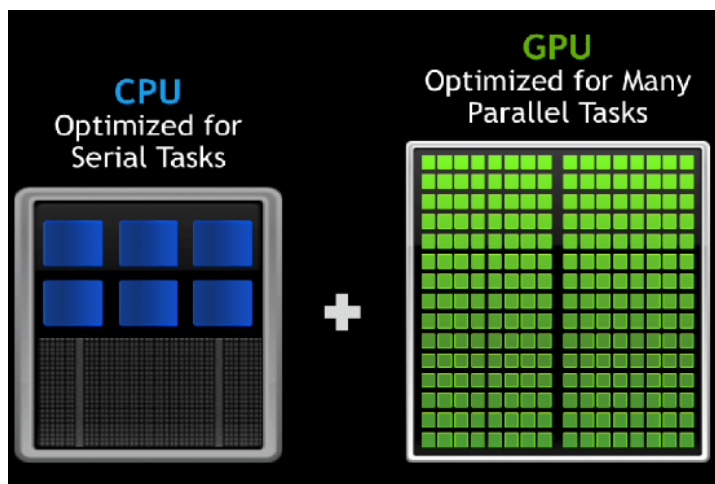
i razvoj namenskog prilagođenog integrisanog kola, koriste se ASIC rešenja. Za razliku od FPGA kola, ovde će izabrani dizajn prilikom razvoja hardvera biti trajno smešten u sam hardver — radi se kompletan postupak fotolitografije i druge faze izrade. Razvijeno ASIC kolo može da sadrži stotine miliona logičkih elemenata, ali i druge komponente kao što su memorije (ROM, RAM, EEPROM, flash) i druge vrste gradivnih blokova. ASIC kola eventualno mogu da sadrže mali deo sa mogućnošću reprogramiranja, ali osnovna ideja je fiksni, optimizovani hardver.

ASIC i FPGA se takođe koriste za računarsku podršku uređaja, slično kao CPU i mikrokontroleri. Oni mogu da realizuju IP jezgra procesora, memorije, komunikacione module i DSP funkcionalnosti. Posebno je zgodno koristiti ASIC kola kada je prihvatljiv visok početni razvojni trošak, a dobijeni namenski proizvod će biti proizveden u velikom broju, čime se amortizuju troškovi i postiže maksimalna efikasnost. Na primer, velike mrežne kompanije koriste namenski ASIC za IP prosleđivanje paketa u ruterima, što daje veliku brzinu i nisku potrošnju. ASIC kola proizvode razne foundry kompanije, među kojima su TSMC, GlobalFoundries, Samsung Semiconductor i druge.

5.3 GPUs

GPU-ovi (Graphics Processing Units) predstavljaju jedinice za obradu grafike. Njihovo ime potiče od početne primene — obrada grafičkih podataka kao što su slike i video. Arhitektura GPU-a optimizovana je za masivnu paralelizaciju: slike i video su prirodno predstavljeni kao matrice piksela, a mnoge operacije se ponavljaju nezavisno za veliki broj piksela ili grupa piksela. Zato GPU poseduje veliki broj paralelnih jedinica (u modernim dizajnim često stotine ili hiljade „jezgara“ ili „threadova“) što omogućava visoku propusnost za operacije koje se lako paralelizuju.

Primene GPU-ova uključuju prikaz grafike na ekranima računara i mobilnih uređaja, renderovanje i procesiranje video signala, ali i domene van klasične grafike: ubrzanje algoritama za mašinsko učenje, neuralne mreže, obradu signala i druge numeričke zadatke koji zahtevaju množenja i sabiranje velikih matrica. Ovo je dovelo do široke primene GPU-ova u autonomnim vozilima, dronovima, sistemima za nadzor, ali i u data-centarima za treniranje modela.



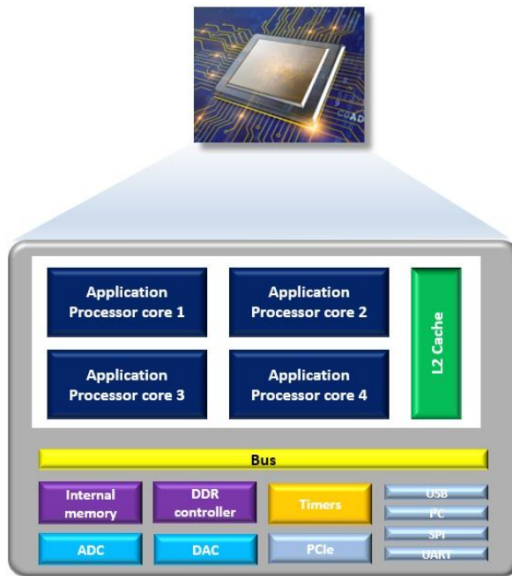
Slika 5.4: Konceptualni prikaz paralelizacije rada na GPU: mnogo niti radi istu operaciju na različitim elementima.

Glavni proizvođači GPU rešenja su NVIDIA, AMD i Intel, a njihova primena se neprestano širi zahvaljujući velikoj efikasnosti u radu sa matricnim operacijama i podršci za akceleraciju AI radnih tokova.

5.4 SoC (System on Chip)

SoC predstavlja integrisani sistem na jednom čipu: u njegovoj srži se nalazi CPU ili mikrokontroler kome su dodate druge hardverske funkcije (memorije, periferije, komunikacioni moduli, akceleratori), sve to smešteno kompaktno na jedinstvenom čipu. Takav pristup donosi bolju pouzdanost sistema, nižu cenu i manju potrošnju energije u poređenju sa diskretnim komponentama i interkonekcijama na ploči.

Dodatne funkcije u SoC-u mogu da uključuju AI/ML akceleratori, hardver za računarski vid, komunikacione module za bežičnu povezanost, kontrolere za kamere i druge funkcionalnosti. Klasičan primer efikasne upotrebe SoC rešenja su mobilni telefoni, gde se na jednom čipu integrišu procesorska jezgra, modemi za mobilne mreže, multimedijalni blokovi i akceleratori za mašinsko učenje.



Slika 5.5: Tipična blok-diagram SoC čipa: CPU, memorija, komunikacioni moduli, AI akceleratori, multimedija.

5.5 Specijalne procesorske jedinice

Pored glavnih procesorskih jedinica, postoje i dodatne specijalizovane jedinice optimizovane za određene klase zadataka. Jedna grupa takvih jedinica su VPU (Vision Processing Units) koje pospešuju rad računarskog vida — prepoznavanje, detekciju i praćenje objekata. Primeri VPU akceleratora su implementacije korišćene u uređajima poput Microsoft HoloLens ili Intel Movidius.

Druga klasa su NNP (Neural Network Processors) odnosno procesori neuronskih mreža, dizajnirani specifično za izvođenje neuralnih mreža efikasno i energetski povoljno. Primeri NNP rešenja uključuju Google Tensor Processing Unit, Intel Nervana, AWS Inferentia i Apple Neural Engine.

Pored NNP, postoje i šire orijentisani procesori za mašinsko učenje (Machine Learning Processors, MLP) koji ciljaju različite algoritme AI/ML, ne samo neuralne mreže. Primeri takvih platformi su ARM Trillium i druge korporativne platforme.

Programabilna kola čine ključni deo modernih pametnih uređaja: FPGA pružaju fleksibilnost i veoma dobru paralelizaciju za specifične

CPU		GPU		TPU	NPU	
						
						
						
						

Slika 5.6: Pregled proizvođača CPU, GPU, TPU i NPU čipova.

zadatke, ASIC nude maksimalnu efikasnost kod velikoserijske proizvodnje, GPU-ovi su idealni za masivne paralelne numeričke operacije (uključujući i AI), a SoC integriše kompletan sistem u kompaktan, energetske efikasan paket. Specijalne procesorske jedinice (VPU, NNP, MLP) dodatno poboljšavaju performanse za računarski vid i mašinsko učenje. Izbor odgovarajuće kombinacije zavisi od zahteva aplikacije, budžeta, očekivanog obima proizvodnje i ograničenja u pogledu energije i latencije.

Deo II

Komunikacije

Glava 6

Hardverska komunikacija upravljačkih računarskih elemenata

Cilj poglavlja

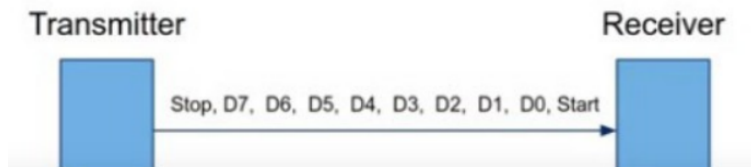
Poglavlje ima za cilj prikaz osnovnih principa serijske i paralelne komunikacije upravljačkih računarskih elemenata sa senzorima i aktuatorima. Objašnjene su karakteristike i primena protokola I²C, I3C i SPI, kao i interpretacija talasnih oblika digitalnih signala. Poseban akcenat stavljen je na probleme visokobrzinskih prenosa i metode njihovog prevazilaženja, omogućavajući razumevanje analize i projektovanja komunikacionih veza između kontrolnih jedinica i periferija.

6.1 Osnove serijske i paralelne komunikacije

6.1.1 Serijska komunikacija

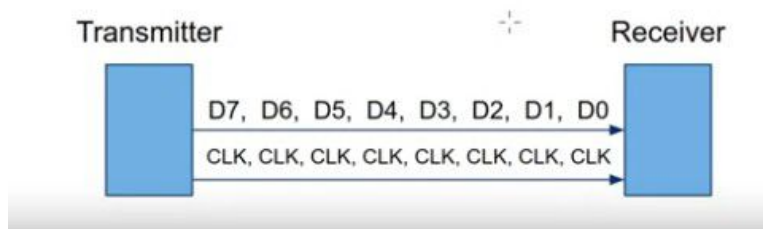
Serijska komunikacija podrazumeva slanje bita jedan po jedan preko žice, koristeći precizan takt vremena. Asinhrona serijska komunikacija postoji veoma dugo. Za ovu vrstu komunikacije koristi se hardverska komponenta označena sa UART (Universal Asynchronous Receiver Transmitter), odnosno univerzalni asinhroni prijem–predajnik. UART se ponekad označava i kao serijski port, RS-232 ili EIA-232.

Prilikom komunikacije predajnik i prijemnik koriste dogovorenu brzinu prenosa (clock speed, odnosno embedded clock), šalju bajt po bajt informacija i koriste start i stop bite za sinhronizaciju.



Slika 6.1: Ilustracija serijske komunikacije.

Neke vrste serijskih protokola zahtevaju dodatne linije za prenos. Na primer, kod I²C koristi se jedna linija za podatke koja se označava SDA (Serial DATA) i jedna linija za takt koja se označava SCL (Serial CLock).



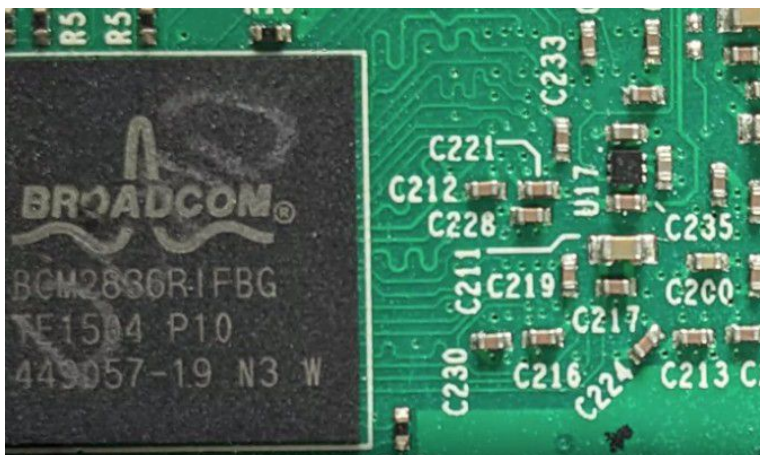
Slika 6.2: Ilustracija I2C serijske komunikacije.

6.1.2 Paralelna komunikacija

Paralelna komunikacija podrazumeva slanje više bita korišćenjem paralelnih žica iste dužine u isto vreme, odnosno tokom jednog ciklusa vremena (clock cycle). Jedan od izazova paralelne komunikacije je obezbeđivanje jednake dužine prenosnih vodova. Da bi se obezbedilo da žice imaju istu dužinu koristi se tehnika koja se označava kao vijuganje (meandering).

Jedan primer vijuganja radi postizanja iste dužine puta za paralelnu komunikaciju je povezivanje između Raspberry Pi Broadcom SoC (System on Chip) i flash memorije na ploči — magistrala velike brzine (high speed data bus).

Drugi primer je vijuganje korišćeno između RPi SoC i kamere, gde se vijuganje primenjuje za diferencijalni signal. Diferencijalni signal



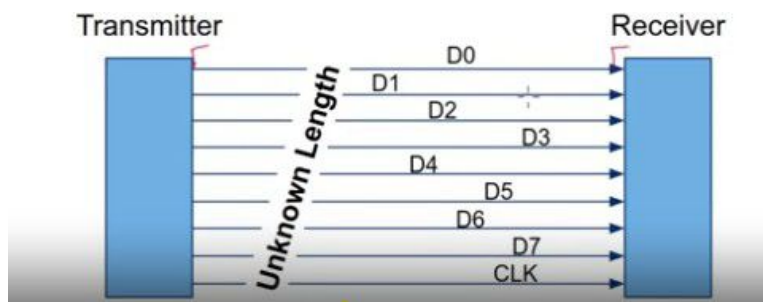
Slika 6.3: Primer vijuganja radi postizanja iste dužine puta signala (SoC – Flash).

predstavlja napon kao razliku susednih linija, što poboljšava otpornost na spoljne smetnje.



Slika 6.4: Primer vijuganja kod diferencijalnog signala (SoC – kamera).

Prilikom paralelnog prenosa velikim brzinama, kada je period bita reda ns, fenomen *skew* (vremenska razlika između dolaska bitova) postaje značajan problem.



Slika 6.5: Ilustracija vremenskih razlika pri prijemu kod paralelne komunikacije velikih brzina (skew).

6.1.3 Tehnološki trendovi u razvoju serijske i paralelne komunikacije

Tokom godina posmatrani su prelazi u smeru između paralelne i serijske komunikacije. Prvi prelazak išao je od serijske ka paralelnoj kako bi se povećala propusnost. Kasnije je usledio povratak ka serijskoj komunikaciji, u naprednim varijantama velike brzine. Paralelna komunikacija je obezbedila veću propusnost, ali po cenu većeg broja vodova, potrebe za preciznim poravnanjem dužina i većeg prostora na pločama.

Novi trendovi uključuju upotrebu više paralelnih serijskih linija (“lanes”) bez zajedničkog takta — na taj način se izbegava problem skew-a koji pogađa klasične paralelne magistrale.

Standard	Tipične brzine	Primer primene
I ² C	do 3.4 MHz (High speed)	povezivanje procesora i periferija
SPI	do ~60 MHz	SD / microSD, ekran, senzori
SATA	do 6 Gbps	hard diskovi / SSD
PCIe	do 64 Gbps single-lane	PCIe dodatne kartice (GPU)

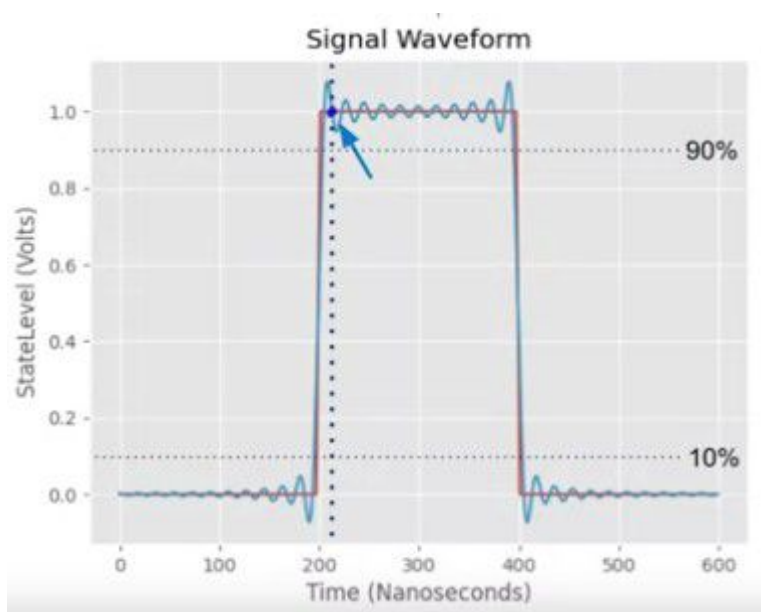
Tabela 6.1: Primeri standarda i tipičnih brzina prenosa.

6.1.4 Talasni oblici digitalnih signala

Logičke vrednosti 1 i 0 predstavljene su električnim signalima; oblik i nivoi tih signala određeni su standardima koji se koriste u specifičnoj komunikaciji. Pri analizi integriteta signala i problema u prenosu važno

je posmatrati vremenske karakteristike (rise/fall times) i pragove za donošenje odluke o logičkoj vrednosti.

Za čitanje talasnog oblika obično se koristi vertikalna isprekidana linija koja označava vreme uzorkovanja. Uobičajeni pragovi su: ispod 30 % V_{DD} za logičku 0 i iznad 70 % V_{DD} za logičku 1. Vreme rasta i padanja signala često se izražava kao procenat (npr. 10 % – 90 %).



Slika 6.6: Primer talasnog oblika digitalnih signala i tipičnih zona za interpretaciju.

Oštećenje signala (npr. zbog smetnji, reverberacije ili nepravilnog poravnanja) može dovesti do pogrešnog očitavanja bita i grešaka u prenosu. Projektovanje visokoperformantnih rešenja uključuje minimizaciju ovih uticaja.

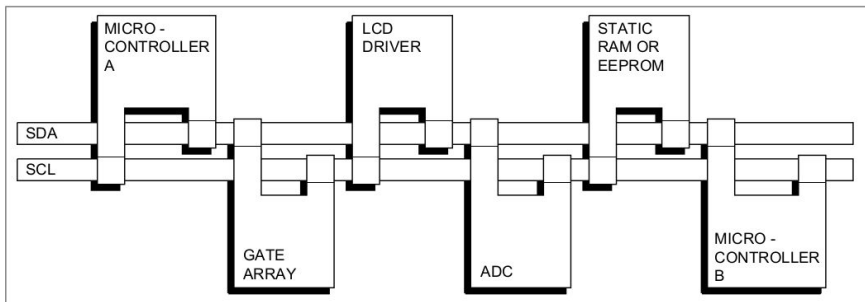
6.2 Serijski protokoli za upravljačke računarske sisteme

6.2.1 I²C komunikacija

I²C (Inter-Integrated Circuit) je prvobitno razvila kompanija Philips (sada NXP) 1982. godine. Specifikacija je dostupna (UM10204

I²C-bus specification and user manual). I²C je dvolinijska serijska magistrala (SDA, SCL), orijentisana na 8-bitne bajtove. Brzine prenosa uključuju: 100 kbps (standard), 400 kbps (fast), 1 Mbps (Fast Plus) i 3.4 Mbps (High Speed).

Jedna magistrala može biti zajednička za mnogo uređaja; SDA i SCL su dvosmerne linije koje se obično povezuju na napajanje preko pull-up otpornika. Uređaji povezuju liniju na 0 V da bi informisali magistralu, dok je normalno stanje HIGH (povučeno ka napajanju).



Slika 6.7: Deljenje I²C magistrale među uređajima.

Logički nivoi (V_{IL} , V_{IH}) su definisani kao procenat od V_{DD} (30 % / 70 %). Validnost podataka zahteva da su podaci stabilni tokom HIGH perioda takta — promene na SDA liniji smeju se dešavati samo dok je SCL LOW.

START i STOP uslovi se definišu preko promena nivoa na SDA dok je SCL HIGH (HIGH→LOW = START, LOW→HIGH = STOP). Svaki bajt prenosa prati bit za potvrdu (ACK); prenos počinje sa MSB.

6.2.2 Uvod u I³C

MIPI I³C (i javno dostupan MIPI I³C Basic) obezbeđuju skalabilnu, srednje brzu, pomonu i kontrolnu magistralu za povezivanje perifernih uređaja sa procesorom aplikacija. Njegov dizajn uključuje ključne attribute i sa I²C-bus i sa SPI interfejsa kako bi se obezbedilo jedinstveno rešenje interfejsa visokih performansi niske potrošnje koje pruža fleksibilnu putanju nadogradnje za I²C-bus i SPI implementatore. I³C predstavlja jedinstveno rešenje za zahteve koje ispunjavaju i I²C i SPI.

I³C može da koegzistira sa I²C u istom kolu. I²C nema direktan način da uređaj pošalje prekid kontroleru. Mora da koristi drugu liniju,

na primer GPIO, za slanje prekida. I3C to može da uradi direktno, a oni se nazivaju in-band interrupt.

Baš kao i I2C, I3C je implementiran sa standardnim CMOS I/O pinovima koristeći dvožični interfejs, ali za razliku od I2C podržava prekide u opsegu, omogućavajući ciljnim uređajima da obaveste kontrolere o prekidima, što je karakteristika dizajna koja eliminiše potrebu za posebnim opšte-namenskim ulaz/izlaz (GPIO) prekidom za svaki cilj, smanjujući troškove i složenost sistema.

I3C podržava multi-drop magistralu koja, na 12,5 MHz, podržava standardnu brzinu prenosa podataka (Standard Data Rate, SDR) od 10 Mbps sa opcijama za režime visoke brzine prenosa podataka (High Data Rate, HDR).

Neto rezultat je da I3C nudi skok u performansama i energetskej efikasnosti u poređenju sa I2C. Besplatna/osnovna verzija I3C uključuje SDR sa brzinom od 12,5 Mbps i HDR-DDR sa brzinom od 25 Mbps.

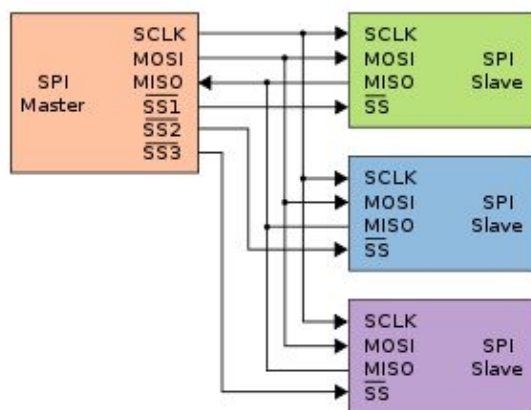
6.2.3 SPI

SPI (Serial Peripheral Interface) je četvorovodični interfejs koji nudi full-duplex prenos i velike brzine (često do nekoliko desetina MHz). Tipične linije su:

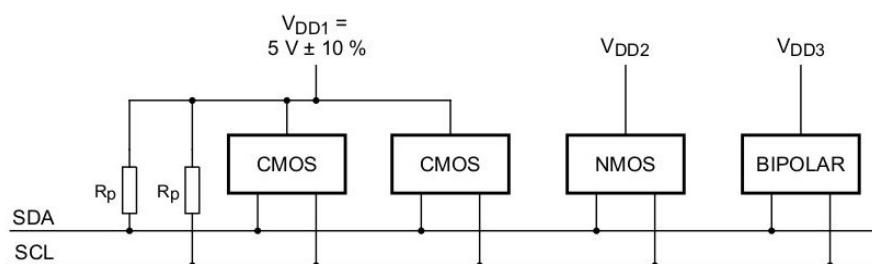
- MOSI (Master Out Slave In) – izlazna linija podataka sa mastera,
- MISO (Master In Slave Out) – ulazna linija podataka u master,
- SCLK (Serial CLock) – taktanu liniju generiše master,
- SS / CS (Slave Select / Chip Select) – linija za izbor slave uređaja.

SPI je jednostavan i brz, ali nema standardizovanu kontrolu toka, niti jedinstven standard (postoje varijante koje menjaju polaritet SCK, aktivnost SS na HIGH umesto LOW itd.). SPI se često koristi za prenos velikih blokova podataka na kratkim rastojanjima (npr. SD kartice, ekrani).

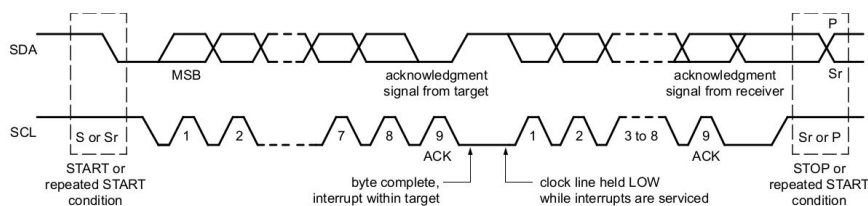
Na slici 6.9 vidi se tipičan primer gde master selektuje jedan od više slave uređaja i razmenjuje podatke u sinhronoj full-duplex komunikaciji. Ne-selektovani slave uređaji drže svoje izlaze u visokoj impedansi kako bi se izbegao konflikt na magistrali.



Slika 6.8: Ilustracija SPI magistrale sa master i slave uređajem.



Slika 6.9: Povezivanje master uređaja sa više slave uređaja na SPI magistrali.



Slika 6.10: Talasni oblici signala na SPI magistrali pri prenosu jednog bajta.

Glava 7

Pregled komunikacionih sistema kratkog i lokalnog dometa za pametne uređaje

Cilj poglavlja

Cilj ovog poglavlja je detaljno prikazati i analizirati komunikacione tehnologije kratkog dometa koje se koriste u pametnim uređajima i IoT sistemima. Poglavlje objašnjava principe rada, arhitekturu, domet, propusnost, energetske karakteristike i tipične primene tehnologija kao što su Bluetooth, BLE, ZigBee, RFID i NFC, Ethernet i Wi-Fi. Poseban fokus je na izboru tehnologije u zavisnosti od zahteva uređaja sa ograničenim resursima, optimizaciji potrošnje energije i integraciji u lokalne i cloud arhitekture.

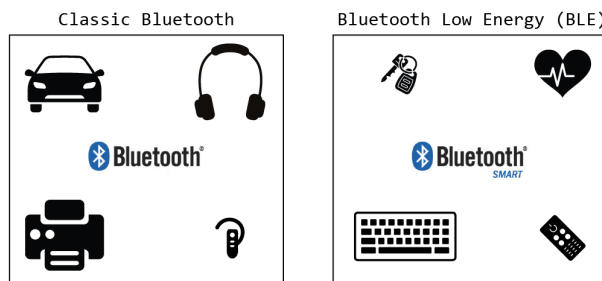
7.1 Bluetooth i Bluetooth Low Energy (BLE)

7.1.1 Uvod u Bluetooth komunikaciju

Bluetooth je bežični komunikacioni standard razvijen za kratkodometnu razmenu podataka između elektronskih uređaja. Inicijalno je razvijen u kompaniji Ericsson, a kasnije standardizovan od strane Bluetooth Special Interest Group (SIG). Funkcioniše u nelicenciranom ISM opsegu na približno 2.4 GHz, koristeći tehniku Frequency Hopping Spread Spectrum (FHSS). Ova tehnika omogućava prebacivanje

komunikacije između različitih frekvencija (do nekoliko hiljada puta u sekundi), što smanjuje verovatnoću smetnji i povećava otpornost na šum i interferenciju.

Uređaji se dele u tri klase prema snazi prenosa i dometu: klasa 1 (domet do približno 100 m, snaga do 100 mW), klasa 2 (domet do približno 10 m, snaga oko 2.5 mW, najčešći u pametnim telefonima) i klasa 3 (domet do približno 1 m, snaga do 1 mW). Komunikacija se zasniva na master-slave arhitekturi. Mreža koja se formira naziva se piconet, a jedan master može imati do sedam aktivnih slave uređaja.

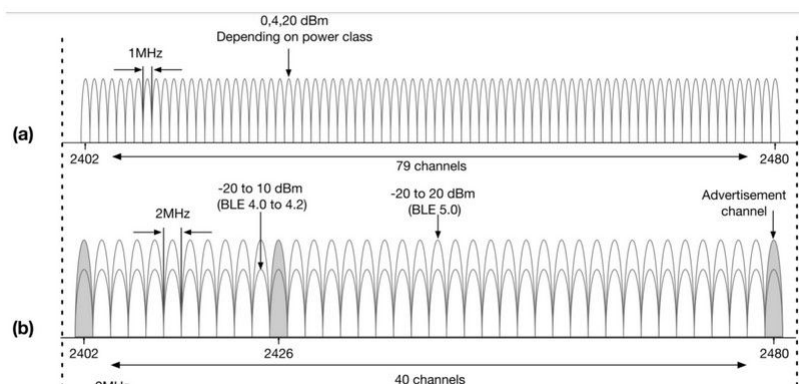


Slika 7.1: Ilustracija osnovne Bluetooth i BLE topologija.

7.1.2 Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) uveden je u verziji Bluetooth 4.0 i predstavlja protokol optimizovan za veoma nisku potrošnju energije. BLE koristi 40 kanala širine 2 MHz, od toga 37 kanala služe za prenos podataka, dok su tri kanala rezervisana za oglašavanje (advertisement channels) sa centralnim frekvencijama 2402 MHz, 2426 MHz i 2480 MHz. Strateškim izborom kanala minimizira se preklapanje sa često korišćenim IEEE 802.11 b/g/n kanalima.

Ključne osobine BLE tehnologije uključuju nisku potrošnju energije, brzo uspostavljanje veze u milisekundnom rasponu, manji protok podataka i jednostavan model razmene informacija zasnovan na GATT (Generic Attribute Profile). BLE oglašavanje (advertising) koristi se za inicijaciju veze. Nakon uspostavljanja, dvosmerna komunikacija se obavlja preko GATT protokola, gde jedan uređaj deluje kao server (npr. senzor), a drugi kao klijent (npr. mobilna aplikacija).



Slika 7.2: Upoređivanje kanala klasičnog Bluetootha i BLE (advertising kanali istaknuti).

7.1.3 Poređenje klasičnog Bluetootha i BLE

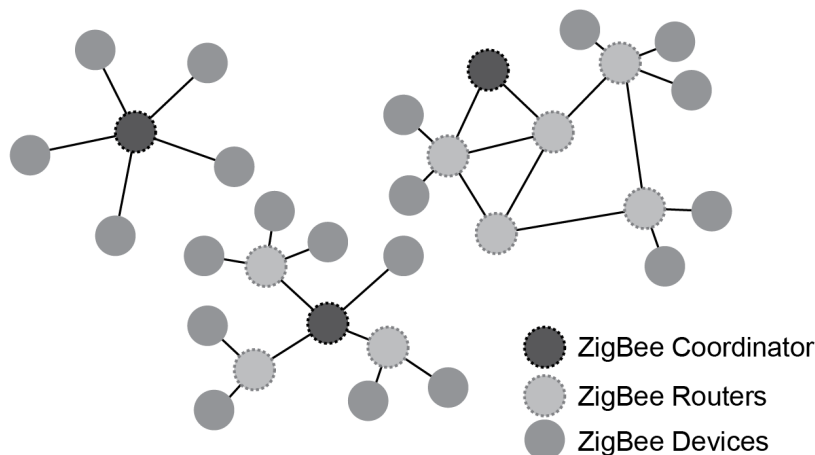
Klasični Bluetooth je primarno optimizovan za kontinuirane podatkovne tokove (audio, fajl transfer), dok je BLE optimizovan za povremene, kratke razmene sa veoma niskom potrošnjom energije. Tipične karakteristike se mogu sažeti ovako: klasični Bluetooth ima veće brzine prenosa (do nekoliko Mbps kod određenih verzija), sporije uspostavljanje veze i veću potrošnju. BLE ima manju brzinu (do 1 Mbps kod BLE 4.x), vrlo brzo uspostavljanje veze i minimalnu potrošnju. Kompatibilnost između uređaja zavisi od podrške za dual-mode implementaciju.

7.2 ZigBee komunikacija

ZigBee je bežična tehnologija za mreže niskog dometa i male potrošnje, zasnovana na IEEE 802.15.4. Osnovna odlika ZigBee mreže je mogućnost formiranja mesh topologije u kojoj se čvorovi međusobno prosleđuju podatke, što povećava pokrivenost i pouzdanost u okruženjima sa velikim brojem uređaja.

Frekvencijski opsezi uključuju 2.4 GHz, 868 MHz i 915 MHz. Maksimalne brzine su relativno male (do 250 kbps u 2.4 GHz opsegu), ali dovoljne za senzorske i upravljačke aplikacije (temperature, pokreta, prekidača, osvetljenja). Arhitektura ZigBee mreže obično uključuje koordinatore, rutere i krajnje uređaje. Takva podela omogućava skalabil-

nost i otpornost na prekid. Prednost ZigBee-a ogleda se u energetskej efikasnosti i pogodnosti za statične mesh mreže, dok su složenost upravljanja mrežom i ograničen kapacitet prenosa faktori koji ga čine manje prisutnim u širokoj potrošačkoj elektronici.



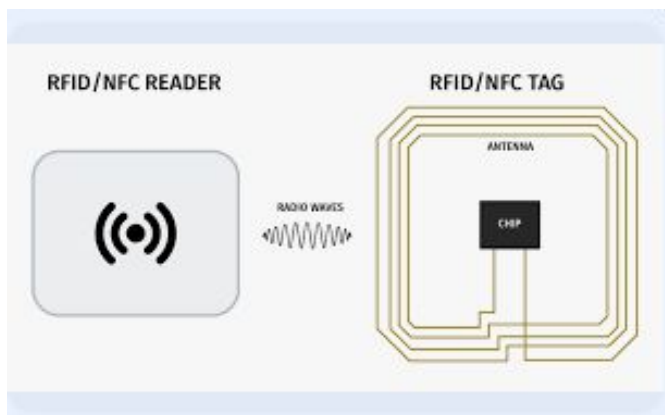
Slika 7.3: Topologije ZigBee mreže.

7.3 RFID i NFC — identifikaciona komunikacija

RFID (Radio-Frequency Identification) i NFC (Near Field Communication) predstavljaju tehnologije za beskontaktnu razmenu podataka na malim rastojanjima, prvenstveno za potrebe identifikacije i autentifikacije. RFID sistem se sastoji od čitača i oznake (tag). Čitač emituje elektromagnetno polje koje aktivira ili napaja oznaku i omogućava razmenu podataka. RFID radi na različitim frekvencijama (npr. 125 kHz, 13.56 MHz, 860–960 MHz) i može koristiti pasivne ili aktivne tagove u zavisnosti od zahteva za dometom i autonomijom.

NFC funkcioniše isključivo na 13.56 MHz i omogućava dvosmernu komunikaciju na vrlo kratkim rastojanjima (do 10 cm). NFC je često ugrađen u mobilne telefone i koristi se za mobilna plaćanja, prenos podataka i autentifikaciju. Ove tehnologije se ne koriste za opšte mrežno

umrežavanje, ali su ključne za scenarije identifikacije korisnika, objekata i događaja (pristupni sistemi, elektronski novčanici, evidencija).



Slika 7.4: Princip komunikacije RFID i NFC tehnologija.

7.4 Ethernet i Wi-Fi komunikacija

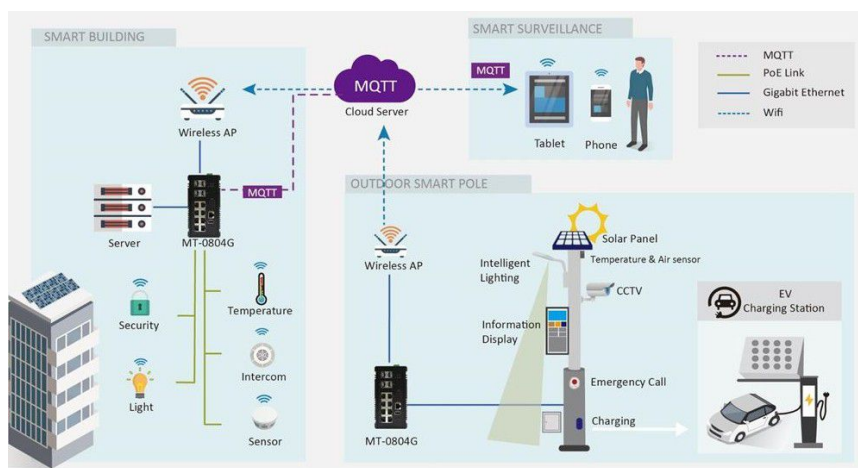
Ethernet i Wi-Fi predstavljaju osnovne tehnologije za žičanu i bežičnu komunikaciju u savremenim sistemima. Razumevanje njihove arhitekture i ograničenja je ključno za projektovanje IoT i pametnih sistema.

7.4.1 Ethernet — žičana komunikacija velikog kapaciteta

Ethernet (IEEE 802.3) obezbeđuje prenos podataka paketima (frejmovima) preko bakarnih ili optičkih medija. Tipične brzine uključuju 10/100 Mbps (Fast Ethernet), 1 Gbps (Gigabit Ethernet) i 10 Gbps. Ethernet se često koristi u fiksnim instalacijama kojima su potrebne mala latencija i visoka pouzdanost. Veliki broj uređaja podržava PoE (Power over Ethernet), čime se istovremeno obezbeđuje napajanje i prenos podataka.

7.4.2 Wi-Fi — bežična fleksibilnost za pametne sisteme

Wi-Fi (IEEE 802.11) omogućava bežični pristup mreži u opsezi-
ma 2.4 GHz i 5 GHz (i 6 GHz u novijim standardima). Tehnike po-
put OFDM (Orthogonal Frequency Division Multiplexing) i CSMA-
/CA (Carrier Sense Multiple Access with Collision Avoidance) koriste
se za efikasno upravljanje deljenim kanalima. Wi-Fi integriše IP sloj,
što omogućava direktnu konekciju na internet i cloud servise. Glavne
prednosti su fleksibilnost i široka rasprostranjenost, a glavni nedostaci
su veća potrošnja energije i osetljivost na radio-smetnje u zagušenim
okruženjima.



Slika 7.5: Primer arhitekture sa lokalnim mrežama: senzori sa niskom potrošnjom komuniciraju sa centralnim čvorom koji koristi Ethernet ili Wi-Fi za pristup internetu.

Glava 8

Mreže sa širim dometom i uporedni pregled tehnologija

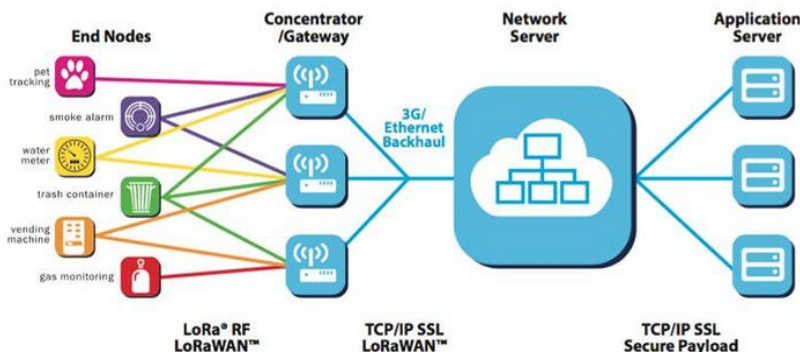
Cilj poglavlja

Cilj ovog poglavlja je detaljno predstaviti komunikacione tehnologije velikog dometa i njihove arhitekture, sa posebnim akcentom na integraciju sa ugrađenim uređajima i IoT sistemima. Poglavlje objašnjava principe rada, prednosti i ograničenja LoRa, NB-IoT i mobilnih mreža, kao i ulogu cloud i edge arhitektura u optimizaciji prenosa podataka, energetske efikasnosti i skalabilnosti sistema. Poseban fokus je na izboru tehnologije u zavisnosti od dometa, potrošnje energije, propusnosti i zahteva aplikacije, uz primenu hibridnih arhitektura za balans između lokalne obrade i centralizovanih servisa.

8.1 LoRa — dug domet uz nisku potrošnju

LoRa je fizička i modulaciona tehnologija zasnovana na chirp spread spectrum modulaciji koja omogućava veoma velik domet uz nisku potrošnju energije. U praksi se postižu udaljenosti reda veličine kilometara, pri čemu je tipičan domet u ruralnim područjima 10–15 km, dok u urbanim sredinama domet može biti znatno manji zbog smetnji i gustine objekata. LoRaWAN definiše mrežni i serverski sloj kojim se uređaji povezuju preko gateway-a ka centralnim network i aplikacionim serverima. Gateway-i koriste standardne kanale (Ethernet, Wi-Fi, mobilne mreže) za dalje prosleđivanje podataka. Ograničenja LoRa si-

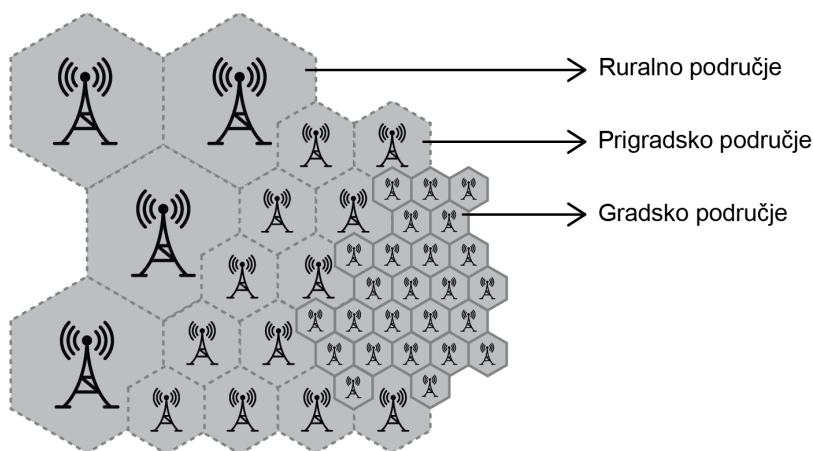
stema proizlaze iz malog kapaciteta prenosa i regulatornih ograničenja u nelicenciranim opsezima, zbog čega je LoRa pogodna za aplikacije sa povremenim slanjem malih poruka, kao što su očitavanja senzora i telemetrija. Prednost je mogućnost izgradnje privatnih mreža i jednostavnost infrastrukturne postavke, dok su mane ograničena propusnost, latencija i zahtevi za planiranjem popunjenosti kanala.



Slika 8.1: Arhitektura LoRaWAN mreže: čvorovi → gateway → network server → aplikacioni server.

8.2 Mobilne telekomunikacije (2G–5G)

Mobilne mreže obuhvataju generacije od 2G do 5G i koriste se kad je potrebna široka pokrivenost, pouzdan IP transport i mobilnost uređaja. Tehnologije poput LTE (4G) i 5G omogućavaju visoke brzine, malu latenciju i napredne QoS mehanizme, što ih čini pogodnim za video prijenos, mobilnu telemetriju i aplikacije koje zahtevaju stalnu konekciju. S druge strane, mobilni moduli obično troše više energije i često zadiru u troškove pretplate i zavisnost od operatora. Zbog toga se za uređaje sa baterijskom autonomijom često vrši kompromis između upotrebljivosti i potrošnje. U mnogim rešenjima mobilne mreže se koriste u kombinaciji sa edge-gateway-ima koji obavljaju lokalnu agregaciju i filtraciju podataka pre slanja u cloud, čime se smanjuje promet i poboljšava energetika celog sistema. Razvoj 5G tehnologija dodatno proširuje mogućnosti za masovno umrežavanje uređaja (mMTC) i scenarije sa vrlo niskom latencijom (URLLC).



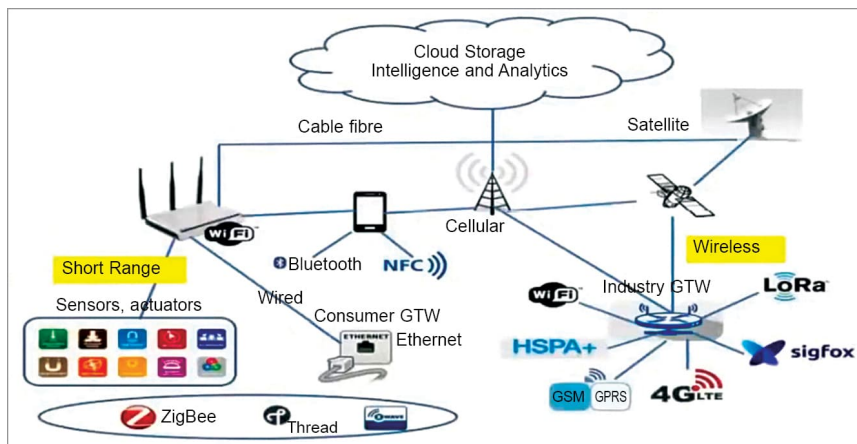
Slika 8.2: Ilustracija tipične mobilne arhitekture za IoT uređaje i gateway konekciju prema internetu.

8.3 NB-IoT — IoT proširenje mobilnih mreža

NB-IoT (Narrowband IoT) predstavlja 3GPP standard osmišljen za masovno povezivanje uređaja sa malim protokom podataka i visokim zahtevima za energetska efikasnost. Tehnologija koristi uski spektar (obično 180 kHz) i omogućava duboku penetraciju signala kroz zidove i podzemne instalacije, što je čini pogodnom za aplikacije kao što su pametna brojila, daljinsko očitavanje i infrastrukturni senzori. Ključne karakteristike NB-IoT su niska potrošnja (PSM, eDRX), velika gustina uređaja po ćeliji i direktna integracija sa mrežom operatora, što olakšava skaliranje i upravljanje uređajima. Za razliku od tehnologija u nelicenciranim opsezima, NB-IoT zahteva saradnju sa mobilnim operatorom i obezbeđuje sigurniji i direktniji put do cloud servisa, uz cenu koja uključuje operatorsku infrastrukturu i potencijalne pretplatničke troškove.

8.4 Cloud i arhitekture za ugrađene uređaje

Arhitektura modernih IoT sistema zasniva se na slojevima: senzorski čvorovi, lokalni gateway/edge sloj i centralizovani cloud servisi. Podaci se često transformišu i agregiraju na gateway-u ili edge-u, nakon čega se selektivno sinhronizuju sa cloud-om radi dugoročne analize

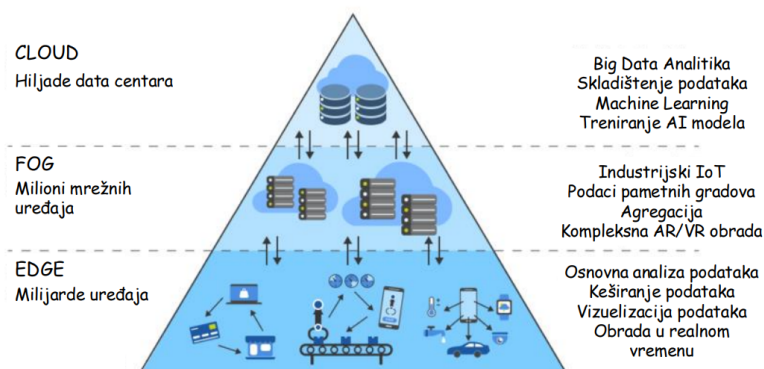


Slika 8.3: Tipična NB-IoT arhitektura: uređaj → mobilna ćelija → operatorova mreža → cloud.

i skladištenja. Na ovaj način se optimizuje upotreba propusnog opsega i smanjuju troškovi prenosa. Edge computing se primenjuje kada su zahtevi za latencijom, privatnošću ili štednjom propusnog opsega stroži — lokalnom obradom se postiže brza reakcija i manja zavisnost od stalne konekcije. Za komunikaciju sa cloud-om obično se primenjuju lagani IoT protokoli kao što su MQTT i CoAP, dok se za sigurnost koriste TLS/DTLS i mehanizmi za upravljanje identitetima i OTA ažuriranja. Kod uređaja sa ograničenim resursima preporučuje se delegiranje teških kriptografskih operacija na gateway ili hardverske bezbednosne module kako bi se smanjilo opterećenje i potrošnja energije.

8.5 Uporedni pregled i implikacije za ugrađene uređaje

Sve predstavljene tehnologije su kontekstualno uporedive kroz kriterijume dometa, propusnosti, potrošnje energije, složenosti mreže i potrebe za infrastrukturom. LoRa i LoRaWAN su pogodniji za privatne, nisko-protok aplikacije sa velikim dometom. NB-IoT nudi robu-snu integraciju preko operatora sa dobrom penetracijom i upravljivim troškovima po uređaju. Mobilne mreže (LTE/5G) pružaju visoke brzine i nisku latenciju ali uz veću potrošnju i troškove povezivanja. Cloud i



Slika 8.4: Hijerarhijska IoT arhitektura: senzorski čvorovi → edge/gateway → cloud servisi.

edge slojevi definišu kako će se podaci agregirati, obrađivati i čuvati. Za ugrađene uređaje izbor tehnologije mora da odrazi kompromis između autonomije baterije, zahteva za pouzdanošću veze i troškova implementacije. U praksi se često primenjuju hibridne arhitekture u kojima periferni senzori koriste niskoenergetske protokole, dok se gateway povezuje putem Ethernet/Wi-Fi ili mobilnih mreža ka cloud-u.

Deo III

Embedded mašinsko učenje

Glava 9

Ograničenja za mašinsko učenje u namenskim računarskim sistemima

Cilj poglavlja

Cilj ovog poglavlja je da detaljno razjasni i analizira ograničenja i izazove sa kojima se susreću mašinsko učenje i njegovo implementiranje u namenskim (embedded) računarskim sistemima. To uključuje hardverske aspekte, kao što su ograničenja procesorske snage, memorije, energetske efikasnosti i specifične arhitekture mikrokontrolera i mikroprocesora, kao i softverske izazove povezane sa ograničenim mogućnostima operativnih sistema, biblioteka i podrške za izvršavanje ML modela na uređajima sa skromnim resursima.

Pored toga, fokusira se i na razvoj i komprimaciju ML modela kroz tehnike poput orezivanja, kvantizacije i destilacije znanja, koje omogućavaju njihovo efikasno korišćenje na ograničenim uređajima. Cilj je omogućiti razumevanje kako se izvođenje ML zadataka sve više prilagođava specifičnostima ugrađenih sistema, kako bi se postigla optimalna balanca između performansi, veličine modela, potrošnje energije i troškova, naročito u kontekstu TinyML pristupa, gde se modeli moraju izvršavati lokalno na uređajima u blizini senzora i aktuatora.

9.1 Uvod

Namenski računarski sistemi (*embedded systems*) obuhvataju i hardver i softver. Glavne komponente hardvera su:

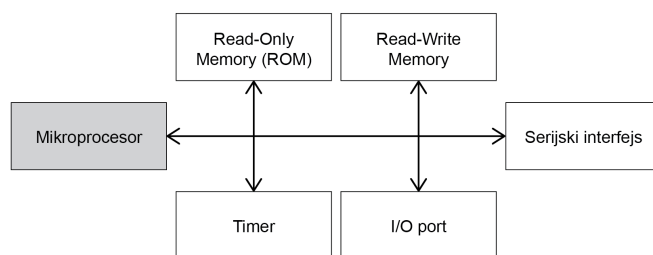
- procesorska jedinica – zadužena za izračunavanje i procesiranje podataka,
- memorija – služi za privremeno skladištenje podataka,
- trajna skladišta podataka – namenjena dugoročnom čuvanju informacija.

9.2 Hardverski segment namenskih računarskih sistema

Procesorski deo sistema može biti zasnovan na različitim arhitekturama, među kojima se najčešće koriste mikroprocesori i mikrokontroleri.

Mikroprocesor predstavlja veću i složeniju komponentu računarske mašine u poređenju sa mikrokontrolerom. Kod mikrokontrolera su procesor, memorija i skladište podataka čvrsto integrisani u jedan monolitni blok, što omogućava kompaktniji dizajn i nižu potrošnju energije.

Način komunikacije i položaj mikroprocesora i mikrokontrolera u odnosu na druge delove sistema prikazani su na sledeće dve slike.

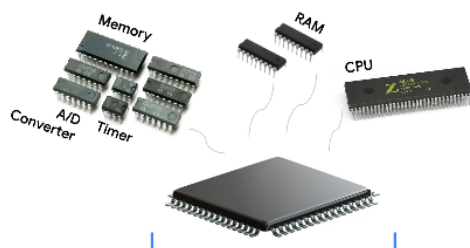


Slika 9.1: Položaj mikroprocesora u odnosu na ostale delove sistema.

Mikroprocesor predstavlja centralni deo računarskog sistema i sastoji se isključivo od procesorske jedinice, dok su memorija i skladištenje podataka eksterni elementi povezani sa njim. Koristi se u sistemima opšte namene, kao što su desktop računari, laptopovi i serveri,

CPU	Read-Only Memory (ROM)	Read-Write Memory
Timer	I/O port	Serijski interfejs

Slika 9.2: Položaj procesorske jedinice u odnosu na druge delove sistema za mikrokontrolere.



Slika 9.3: Mikrokontroler kao kompaktan sistem različitih celina.

gde je potrebna velika fleksibilnost u dizajnu i mogućnost proširivanja funkcionalnosti. Zbog toga što zahteva dodatne komponente, ukupna veličina sistema zasnovanog na mikroprocesoru je relativno velika.

Sa druge strane, mikrokontroler je srce namenskog računarskog (embedded) sistema, kod koga su procesor, memorija i skladištenje integrisani u okviru jednog kompaktnog čipa. On je namenjen specifičnim funkcijama u sistemima kao što su mobilni telefoni, multimedijalni plejeri, kućni aparati i brojni drugi uređaji sa jasno definisanom namenom. Mikrokontroler je ograničen u pogledu fleksibilnosti dizajna, ali zato zauzima znatno manju fizičku veličinu i omogućava ekonomičniju i energetski efikasniju realizaciju sistema.

Generalni pregled brojčanih parametara koje karakterišu mikroprocesor i mikrokontroler dati su u sledećoj Tabeli 9.1.

Ovakve brojčane vrednosti u pogledu brzine rada procesora mikrokontrolera dalje ograničavaju i rad i rešavanje složenih zadataka i u pogledu mogućnosti i u pogledu brzine rada. Sa druge strane i brojčane vrednosti za memoriju određuje i složenost i brzinu rada u slučaju kompleksnih operacija.

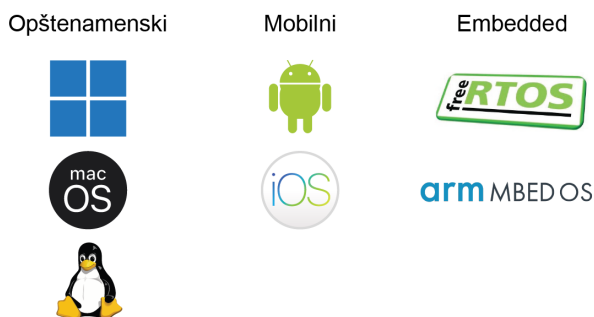
Tabela 9.1: Upporedne brojčane karakteristike mikroprocesora i mikrokontrolera.

Karakteristika	Mikroprocesor	Mikrokontroler
Takt procesora	1–5 GHz	1–200 MHz
Radna memorija (RAM)	1–64 GB	1–512 KB
Programska memorija (ROM / Flash)	256 MB – 4 TB	4 KB – 2 MB
Broj jezgara	2–16	1–2
Potrošnja energije	10–150 W	0.1–1 W
Cena po jedinici	20–500 EUR	0.5–20 EUR
Površina čipa	100–500 mm ²	1–50 mm ²
Tipične primene	Desktop, laptop, serveri	IoT, senzori, kućni uređaji

9.3 Izazovi u pogledu softvera embedded sistema

Sofverska strana hardverskih sistema može se posmatrati u pogledu slojeva koje čine operativni sistem, biblioteke i aplikacije.

Najčešće korišćeni operativni sistemi razvrstani su u tri kategorije: opštenamenski operativni sistemi, sistemi za mobilne uređaje i operativni sistemi namenjeni ugrađenim (embedded) uređajima.



Slika 9.4: Operativni sistemi: opštenamenski, mobilni i embedded.

Namenski računarski sistemi nemaju široku mrežu operativnih sistema. Ovo je posledica toga što je njihova namena specijalizovana, čime se gubi potreba za platformom opšte namene. Ovo ne znači da namenski računarski sistemi uopšte nemaju operativne sisteme. Primeri operativnih sistema za embedded sisteme su *FreeRTOS* i *Arm Mbed OS*.

Međutim, generalno ne postoji tendencija da se oni instaliraju, jer bilo koje dodavanje na ugrađeni sistem oduzima njegove resurse, kao što su memorija i procesorska snaga, čime se smanjuju mogućnosti same aplikacije.

Sledeći viši nivo u pogledu softvera čine biblioteke. Kada na nižem nivou postoji operativni sistem, on obezbeđuje visok nivo interoperabilnosti biblioteka i aplikacija između različitih uređaja, bez obzira na tip procesora (Intel, AMD) koji se koristi. To je jedna od očekivanih osobina sistema opšte namene – isti kod se izvršava na isti način.

Međutim, kada se pređe na ugrađene sisteme, vrednosti i operacije mogu biti značajno različite. Na primer, ako posmatramo iracionalan broj π , koji se može približno izraziti kao $22/7$, imamo broj sa decimalnim zarezmom i velikim brojem cifara iza zareza. U računarskim sistemima vrednost sa pomičnim zarezmom (*floating-point number*) obično se izražava kao tri komponente: znak, eksponent i mantisa (razlomak). Ovi brojevi mogu biti sa jednostrukom (32 bita) ili dvostrukom preciznošću (64 bita). To znači da se i operacije izvršavaju prema pravilima odgovarajućeg formata. U ugrađenim sistemima, broj bita je manji, pa je i preciznost smanjena.

U računarskim sistemima se nad brojevima u određenom formatu najčešće izvršavaju operacije kao što su sabiranje, množenje i deljenje. Ako su brojevi tipa *floating-point*, a implementacija realizovana samo softverski, ovaj postupak je spor i usporava rad sistema. Zbog toga su vremenom razvijene hardverske jedinice za obradu brojeva sa pomičnim zarezmom — FPU (*Floating-Point Unit*). Ova jedinica je danas uobičajena u procesorima opšte namene, kao što su Intel procesori. Iako FPU ubrzava rad i generalizuje kod, ona zahteva dodatno napajanje, fizički prostor i povećava cenu uređaja.

Ove osobine su ograničavajuće za namenski ugrađene sisteme, čija je optimizacija usmerena na nižu cenu, veću energetska efikasnost i specifičnost primene. U mnogim slučajevima FPU nije prisutna, pa kod pisan za jedan sistem često nije prenosiv na drugi. Ako, na primer, razvijen ML model radi na uređaju sa FPU jedinicom, njegovo preba-

civanje na novi namenski sistem bez FPU može dovesti do ozbiljnih problema.

Posmatrajući sisteme sa mikroprocesorima (opšte namene) i sisteme sa mikrokontrolerima (namenske), uočava se da se univerzalnost softverskog dela značajno smanjuje prelaskom na namenski sistem. Čak i kod popularnih platformi kao što su *Raspberry Pi* i *Arduino*, postoji specifična hardverska implementacija biblioteka zavisno od konkretnog uređaja. Dodavanje mašinskog učenja na ovakve sisteme, koji nemaju potpunu portabilnost softvera, postaje dodatni izazov.

Sa druge strane, namenski sistemi omogućavaju manju veličinu uređaja, nižu cenu i manji razvojni tim. Pored toga, efikasnost sistema u pogledu energije i procesiranja često je znatno veća.

Embedded hardver je, dakle, ekstremno ograničen u performansama, memoriji i potrošnji energije, ali upravo to ga čini pogodnim za TinyML primene — pod uslovom da se softver pažljivo optimizuje za njegove mogućnosti.

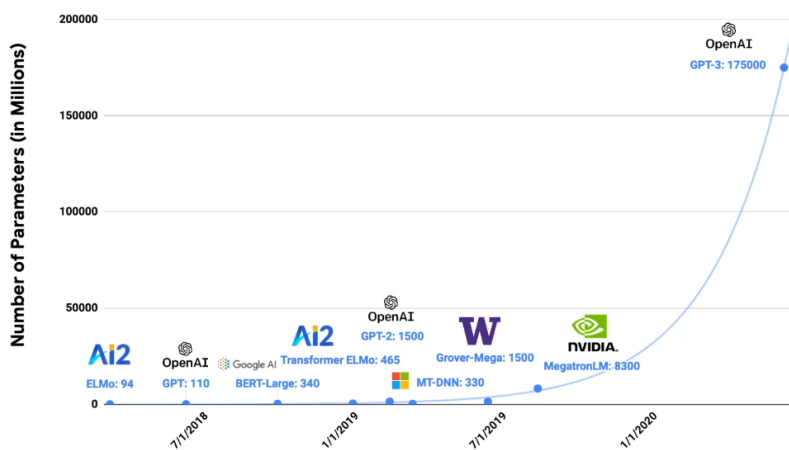
9.4 Izazovi ML algoritama za TinyML

Modeli koji se koriste za rešavanje zadataka mašinskog učenja (ML) postaju sve napredniji u svojim mogućnostima, ali istovremeno i sve složeniji u pogledu broja parametara, memorijskih zahteva i potrebnih performansi računarskog sistema. Time raste i zahtev za procesorskom snagom i resursima neophodnim za njihov rad.

Kompleksnost modela raste izuzetno brzo, a performanse hardverskih sistema moraju pratiti taj rast. Od 2012. godine, kada je započeo veliki razvoj mašinskog učenja zahvaljujući AlexNet-u i upotrebi GPU-ova, računarske potrebe porasle su i do 300.000 puta. U početku, rast mašinskog učenja od 1960-ih bio je linearan, ali se u poslednjoj deceniji razvio u eksponencijalni trend.

Kako bi se obezbedila dovoljna procesorska snaga za ovakve zadatke, razvijeni su namenski procesori za ML, kao što su Google-ovi TPU (*Tensor Processing Unit*) čipovi. Ove jedinice su smeštene u data centre, često raspoređene u ormare (*rack-ove*) u velikom broju, uz napredne sisteme hlađenja.

Ovakvi centri troše ogromne količine energije i često su locirani pored izvora obnovljive energije, poput vetroelektrana ili hidrocentrala. Google-ov centar u Holandiji je dobar primer ove prakse.



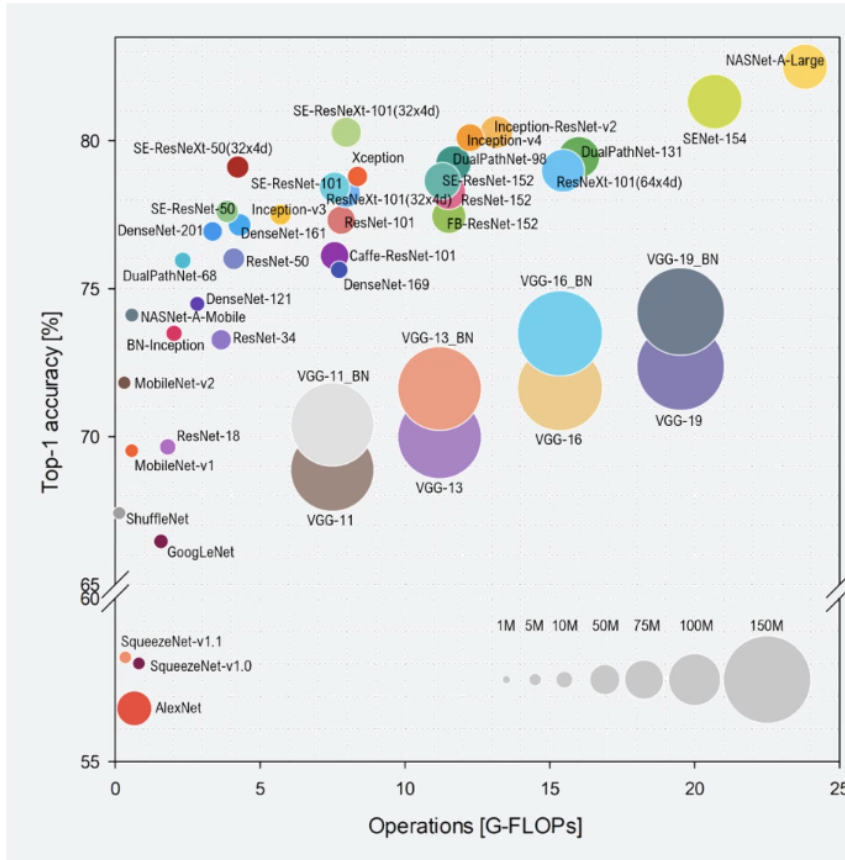
Slika 9.5: Veličina NLP modela izražena preko broja parametara tokom godina.



Slika 9.6: Raspored TPU jedinica u okviru data centra.

9.4.1 TinyML i namenski procesori

Za razliku od data centara, TinyML sistemi imaju potpuno suprotne izazove: minimalne resurse i potrebu za efikasnim izvođenjem modela na uređajima u blizini senzora. Dok ML u oblaku koristi masivne TPU klastere, TinyML zahteva male, energetski štedljive procesore i memorijski optimizovane modele.



Slika 9.7: Evolucija ML modela u pogledu broja operacija, tačnosti i broja parametara.

Na slici 9.7 prikazana je zavisnost između složenosti modela, tačnosti i potrebnih računarskih resursa. Na x-osi prikazan je broj operacija, na y-osi tačnost modela, dok veličina kruga pokazuje memorijski zahtev modela.

Kao što se vidi, postoji jasna tendencija ka smanjenju veličine modela uz zadržavanje visoke tačnosti. Model MobileNet-v1 je dobar pri-

Tabela 9.2: Uporedne karakteristike ML modela.

Model	Godina	Tačnost	Veličina modela
AlexNet (Image-Net, 100 klasa)	2012	57.1%	61 MB
VGGNet	2014	71.5%	528 MB
ResNet (Micro-soft)	2015	75.8%	100 MB
MobileNet-v1	2015	70.6%	16.9 MB

mer – iako zauzima samo 16,9 MB, postiže veoma dobre performanse.

Međutim, kada uporedimo ovu veličinu sa, na primer, Arduino BLE Sense mikrokontrolerom koji ima samo 256 KB RAM memorije, postaje jasno koliko je izazovno smeštanje ML modela na male uređaje. Ugrađeni mikrokontroleri raspolažu kilobajtima memorije, dok su ML modeli u megabajtima.

9.4.2 Kompresija ML modela

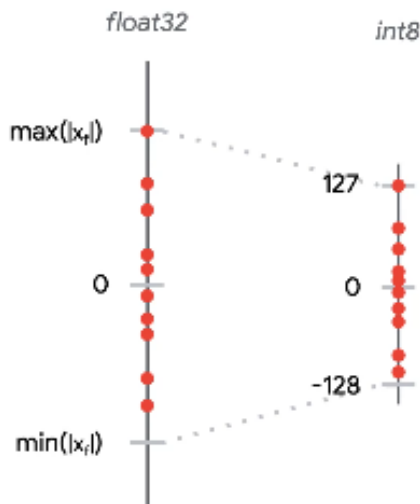
Da bi se modeli mogli da se koriste na mikrokontrolerima, primenjuju se tehnike kompresije, uključujući *pruning* i *quantization*.

Orezivanje (pruning) Orezivanje podrazumeva brisanje određenih neuronskih veza ili neurona koji imaju mali doprinos učenja. Na taj način se smanjuje broj parametara i veličina mreže, dok se tačnost u velikoj meri zadržava.



Slika 9.8: Ilustracija kompresije ML modela tehnikom orezivanja.

Kvantizacija (quantization) Kvantizacija se zasniva na smanjenju broja bita potrebnih za prikaz brojeva. Na primer, prelaskom sa 32-bitnog prikaza (float32) na 8-bitni (int8), veličina modela se može smanjiti i do četiri puta, uz minimalan gubitak preciznosti. Ova metoda je naročito korisna za mikrokontrolere koji nemaju FPU.



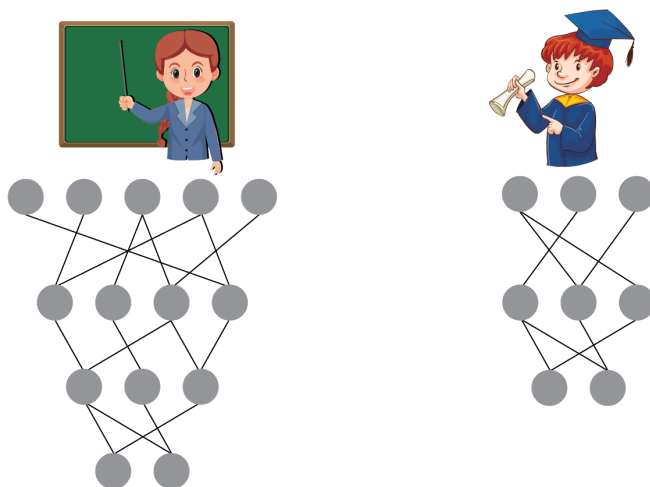
Slika 9.9: Princip kvantizacije u prikazu brojeva.

Destilacija znanja (knowledge distillation) Ova metoda podrazumeva da velika i složena mreža (učitelj) prenosi znanje manjoj mreži (učeniku), zadržavajući ključne informacije uz znatno manju kompleksnost.

9.4.3 Softverski okviri za izvršavanje ML modela

Niz TPU jedinica u okviru data centra i oblaka namenjen je za izvršavanje velikog broja različitih ML zadataka. Za rad sa ovim sistemima potreban je odgovarajući softverski okvir (*software framework*). Jedan od najpoznatijih i najčešće korišćenih softverskih okvira razvijen od strane kompanije Google je *TensorFlow*.

Sa druge strane, ugrađeni namenski procesori imaju znatno manje memorije i procesorske moći, te se ne mogu koristiti za kompletan razvoj i treniranje ML modela. Umesto toga, mogu da se koriste samo



Slika 9.10: Destilacija znanja.

za fazu zaključivanja (*inference*), dok se proces učenja i prepoznavanja obrazaca obavlja u oblaku.

Za proces učenja je potrebno mnogo vremena i resursa, dok je zaključivanje, kada je model već istreniran, znatno brže i efikasnije. Isti princip primenjuje se i u sistemima sa ograničenim resursima.

Zbog toga, umesto punog TensorFlow okvira, za sisteme sa manjim mogućnostima u pogledu procesiranja koristi se softverski okvir *TensorFlow Lite*. TensorFlow je namenjen ML istraživačima koji rade na razvoju i treniranju ML algoritama za rešavanje određenog problema, dok je TensorFlow Lite usmeren na razvoj aplikacija koje integrišu već istrenirane ML modele u proizvode krajnjih korisnika. To predstavlja suštinsku razliku između ova dva pristupa.

Topologija neuronske mreže opisuje način povezivanja između neurona. U okviru TensorFlow okvira, ove veze se tokom procesa treniranja dinamički menjaju i prilagođavaju, dok su u TensorFlow Lite verziji one fiksirane, zajedno sa vrednostima težina (*weights*).

Kada je model u TensorFlow-u razvijen i zadovoljavajuće funkcioniše, moguće ga je konvertovati pomoću *TensorFlow Lite Converter* alata u datoteku sa ekstenzijom `.tflite`. Ova konverzija rezultuje modelom manje veličine i sa fiksnim parametrima, spremnim za upotrebu na ugrađenim računarskim sistemima, poput mobilnih telefona ili spe-

cijalizovanih uređaja sa ograničenim resursima.

Dakle, tok rada uključuje razvoj modela u TensorFlow-u, zatim konverziju, optimizaciju i korišćenje modela uz pomoć TensorFlow Lite okvira. Za primenu na mikrokontrolerima potreban je dodatni sloj za izvršavanje (*TF Lite engine*), a za još manja okruženja koristi se varijanta *TensorFlow Lite for Microcontrollers* (*TensorFlow Micro*).

U okruženju data centara koristi se *TensorFlow*, razvijen od strane Google-a, koji obezbeđuje razvoj, treniranje i testiranje ML modela. Uređaji sa ograničenim resursima ne mogu podržati ceo razvojni proces, pa se na njima koriste modeli koji su već trenirani, a fokus je na njihovom izvršavanju (*inference*).

Za ovakve sisteme razvijen je *TensorFlow Lite*, koji omogućava optimizovano izvršavanje modela na mobilnim i ugrađenim uređajima. Još kompaktnija verzija namenjena mikrokontrolerima je *TensorFlow Lite for Microcontrollers*.

Tok rada obuhvata razvoj i treniranje modela u TensorFlow okruženju, zatim konverziju pomoću *TensorFlow Lite Converter*, optimizaciju i korišćenje na ciljnom uređaju sa ograničenim resursima. Na taj način omogućava se primena ML modela čak i na mikrokontrolerima.

Topologija mreže, vrednosti težina (*weights*), veličina modela i zahtevi za procesiranjem značajno se razlikuju između *TensorFlow* i *TensorFlow Lite* okvira. U *TensorFlow* okruženju topologija mreže je fleksibilna i dinamična tokom treniranja, dok su težine promenljive i model može biti veoma velik, što često zahteva distribuirano procesiranje na klasterima GPU-a ili TPU-a. Zbog toga, razvoj u *TensorFlow*-u je namenjen ML istraživačima i inženjerima sa dubokim znanjem mašinskog učenja i računarstva visokih performansi. S druge strane, *TensorFlow Lite* koristi fiksnu topologiju mreže i unapred određene težine, dok je binarna veličina modela znatno smanjena primenom optimizacija poput kvantizacije i orezivanja (*pruning*). Ovi modeli su dizajnirani da rade na uređajima sa ograničenim resursima, često bez FPU jedinica, i omogućavaju izvođenje zaključivanja (*inference*) lokalno, bez potrebe za distribuiranim sistemima. Shodno tome, *TensorFlow Lite* je primarno namenjen aplikativnim developerima koji integrišu ML modele u krajnje proizvode, dok kompleksno treniranje ostaje u domenu *TensorFlow* okruženja.

Glava 10

Osnove mašinskog učenja

Cilj poglavlja

Cilj ovog poglavlja je da se pruži pregled osnovnih principa mašinskog učenja i neuronskih mreža. Posebna pažnja posvećena je razlikama između tradicionalnog, eksplicitnog kodovanja i pristupa zasnovanog na učenju iz podataka. U okviru poglavlja objašnjavaju se proces učenja i zaključivanja (*inference*), načini merenja tačnosti modela i optimizacije parametara primenom gradijentnog pada, kao i osnovne komponente neuronskih mreža, uključujući slojeve, težine i odstupanja. Dodatno se razmatraju specijalizovani tipovi slojeva, kao što su konvolucijski i rekurzivni slojevi, i njihova primena u obradi različitih tipova podataka. Na kraju se opisuju principi inicijalizacije parametara, propagacije unazad (*backpropagation*) i iterativnog prilagođavanja modela radi minimizacije funkcije greške.

10.1 Eksplicitno i ML kodovanje

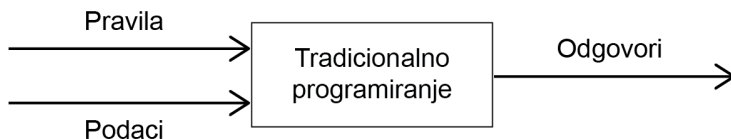
Tradicionalni način programiranja podrazumeva *eksplicitno kodovanje*. U ovom pristupu programer definiše promenljive i skup pravila po kojima se one menjaju. Sve promene u programu su unapred definisane i determinističke, što znači da za iste ulazne podatke program uvek proizvodi isti rezultat.

Na primer, pseudokod koji klasifikuje vrste kretanja može definisati parametar brzine: male vrednosti označavaju hodanje, srednje vrednosti trčanje, a veće vožnju bicikla. Ovakav pristup je jednostavan za

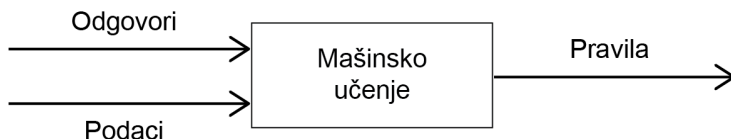
implementaciju i testiranje, ali postaje neprecizan kada se susretnemo sa neočekivanim situacijama, poput igranja golfa ili kombinovanih aktivnosti. Njegova prednost je potpuna kontrola nad logikom programa i deterministički rezultati, dok su mane niska fleksibilnost i teško skaliranje na kompleksne ili nepredvidive situacije.

Paradigma *mašinskog učenja* (*Machine Learning, ML*) uvodi drugačiji pristup. Umesto da programer eksplicitno definiše pravila, ML model uči iz podataka. Na osnovu primera, koji uključuju ulazne podatke i odgovarajuće oznake (*labels*), algoritam sam otkriva pravila i koristi ih za prepoznavanje obrazaca u nepoznatim podacima. Ovaj proces se naziva *zaključivanje* (*inference*).

Prednost ML pristupa je u njegovoj fleksibilnosti i sposobnosti da uči iz podataka, omogućavajući primenu na kompleksne i nepredvidive situacije. Njegove slabosti uključuju potrebu za većom količinom podataka za učenje, složeniju implementaciju i optimizaciju, kao i moguće nepredvidive rezultate u nekim situacijama.



Slika 10.1: Eksplicitno kodovanje: programer definiše pravila i logiku sistema.



Slika 10.2: Pristup mašinskog učenja: model sam uči pravila iz podataka i koristi ih za zaključivanje.

Proces učenja u ML sistemu započinje inicijalnom procenom ili slučajnim postavljanjem vrednosti. Nakon toga se meri tačnost modela na podacima koji imaju odgovore, i vrednosti se optimizuju. Ovaj

ciklus se ponavlja sve dok model ne dostigne zadovoljavajuću preciznost, omogućavajući sistemu da samostalno otkriva pravila koja su neophodna za klasifikaciju ili predikciju novih podataka.

10.2 Merenje tačnosti i gradijentni pad

Jedan od osnovnih zadataka u mašinskom učenju jeste merenje tačnosti modela, odnosno procena koliko dobro model predviđa rezultate na osnovu podataka. Jedna od često korišćenih mera za regresione zadatke je srednja kvadratna greška (*Mean Squared Error, MSE*). Ova mera pokazuje prosečnu kvadratnu odstupanja između stvarnih i predviđenih vrednosti, i predstavlja jednostavan, ali vrlo ilustrativan primer principa merjenja tačnosti.

U praktičnom primeru, možemo definisati linearni model sa parametrima w i b , koji generiše predviđene vrednosti $Y = wX + b$. Zatim se izračunava kvadratna razlika između stvarnih vrednosti i vrednosti predviđenih modelom. Suma svih kvadratnih grešaka, odnosno MSE, daje uvid u ukupnu tačnost modela. Eksperimentisanjem sa različitim parametrima moguće je pratiti kako se promena težina i praga (w i b) odražava na gubitak, i vizualizovati gde se model najpreciznije poklapa sa podacima.

Algoritam 1 Izračunavanje srednje kvadratne greške

- 1: Inicijalizuj parametre modela w i b
 - 2: Za svaki ulaz x_i :
 - 3: Izračunaj predviđenu vrednost $y_i = w \cdot x_i + b$
 - 4: Izračunaj kvadratnu grešku: $(y_i - \hat{y}_i)^2$
 - 5: Saberite sve kvadratne greške da biste dobili ukupnu grešku
 - 6: Po želji, uzmite kvadratni koren za konačni gubitak
-

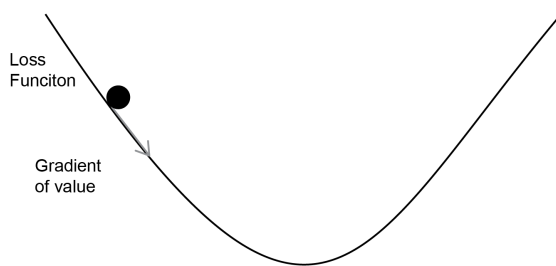
Listing 10.1: Izračunavanje srednje kvadratne greške (MSE)

```

1 import math
2
3 # Parametri modela
4 w = 5
5 b = -2
6
7 # Ulazne i stvarne vrednosti
```

```
8 x = [-1, 4, 1, 0, 1, 4]
9 y_true = [-3, -1, 1, 3, 5, 7]
10
11 # Izračunavanje predikcija
12 y_pred = [w * xi + b for xi in x]
13
14 # Izračunavanje ukupne kvadratne greške
15 total_square_error = sum((y_t - y_p)**2 for y_t, y_p
16                           in zip(y_true, y_pred))
17 loss = math.sqrt(total_square_error)
18
19 print("Stvarne vrednosti Y:", y_true)
20 print("Predvidjene vrednosti Y:", y_pred)
21 print("Srednja kvadratna greška (MSE):", loss)
```

Određivanje gradijenata i izvoda funkcije gubitka predstavlja ključan korak u optimizaciji modela. Funkcija gubitka, kao što je MSE, često ima parboličan oblik, pri čemu minimum funkcije odgovara najmanjem mogućem gubitku. Početna procena parametara modela može se nalaziti bilo gde na paraboli, ali određivanjem gradijenta funkcije u toj tački možemo saznati smer u kojem se minimum nalazi.

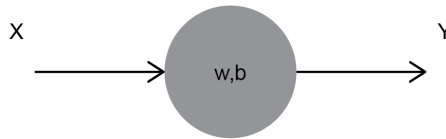


Slika 10.3: Parabolični oblik funkcije gubitka i smer gradijentnog pada ka minimumu.

Velicina koraka kojim se krećemo duž tog smera označava se kao brzina učenja (*learning rate*). Ako je korak premali, konvergencija do minimuma traje dugo, dok preveliki korak može dovesti do preskakanja minimuma. Nakon svakog koraka, izračunava se novi gradijent koji daje novi smer kretanja ka minimalnoj vrednosti funkcije gubitka. Ovaj postupak se ponavlja iterativno sve dok se ne postigne optimalna kombinacija parametara modela, odnosno minimalna vrednost gubitka.

10.3 Uvod u neuronske mreže

Neuronske mreže predstavljaju osnovu savremenog mašinskog učenja i veštačke inteligencije. One funkcionišu kao matematički modeli inspirisani biološkim neuronima, gde svaka jedinica kombinuje ulazne podatke sa ponderima, dodaje odstupanje i kroz funkciju aktivacije generiše izlaz. Parametri pondera (*weights*) i odstupanja (*bias*) služe za određivanje jačine i pomaka signala, a njihovo podešavanje omogućava mreži da uči obrasce iz podataka. Podaci prolaze kroz slojeve, od ulaznog, preko skrivenih, do izlaznog sloja, pri čemu se svaki neuron u sloju povezuje sa svim neuronima u sledećem sloju, što se označava kao gusta (dense) veza. Kada se podaci procesuiraju sloj po sloj, mreža je sekvencijalna.



Slika 10.4: Struktura veštačkog neurona: ulazi x_i , težine w_i , suma i funkcija aktivacije.

Proces učenja zasniva se na minimizaciji funkcije greške, što se postiže primenom gradijentnog pada. Funkcija gubitka, kao što je srednja kvadratna greška (*Mean Squared Error, MSE*), meri odstupanje između stvarnih i predviđenih vrednosti. Optimizator, na primer stohastički gradijentni pad (*SGD*), koristi gradijente funkcije gubitka kako bi odredio smer i veličinu promena parametara. Veličina koraka u ovom procesu, poznata kao brzina učenja (*learning rate*), direktno utiče na brzinu i stabilnost konvergencije ka minimalnoj grešci.

Kratak primer implementacije neuronske mreže sa jednim neuron omogućava ilustraciju ovih principa:

Listing 10.2: Jednostavna neuronska mreža u *Keras/TensorFlow*

```
1 import tensorflow as tf
2 import numpy as np
3
4 model = tf.keras.Sequential([tf.keras.layers.Dense(
    units=1, input_shape=[1])])
```

```
5 model.compile(optimizer='sgd', loss='  
    mean_squared_error')  
6  
7 xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype  
    =float)  
8 ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0],  
    dtype=float)  
9  
10 model.fit(xs, ys, epochs=500)
```

U ovom primeru, parametar `units=1` označava da sloj sadrži jedan neuron, dok `input_shape=[1]` definiše da mreža prima jedan ulaz. Optimizator `sgd` primenjuje stohastički gradijentni pad kako bi se težine i odstupanje prilagodili, dok funkcija gubitka `mean_squared_error` meri kvadratne razlike između predviđenih i stvarnih vrednosti. Argument `epochs=500` označava broj prolaza kroz kompletan skup trening podataka, pri čemu se parametri iterativno ažuriraju u skladu sa gradijentima.

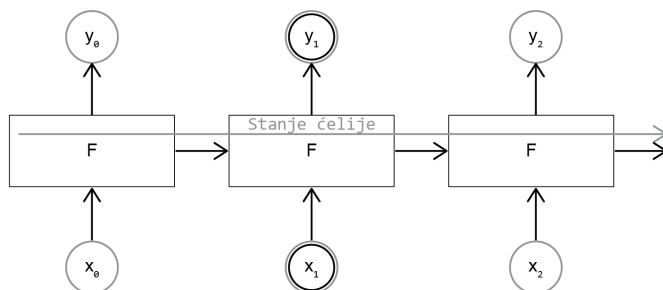
Nakon treniranja, mreža može da izvrši predikciju na nepoznatim podacima (*inference*). Naučena težina povezuje ulaz sa izlazom linearно, dok odstupanje omogućava pomak funkcije. U složenijim mrežama kombinacije više neurona i slojeva omogućavaju modelovanje nelinearnih odnosa i prepoznavanje složenih obrazaca u podacima. Svaki neuron u mreži poseduje sopstvene parametre, koji se iterativno prilagođavaju tokom učenja, čime se omogućava generalizacija na nove, nepoznate ulaze.

10.4 Vrste slojeva u neuronskim mrežama

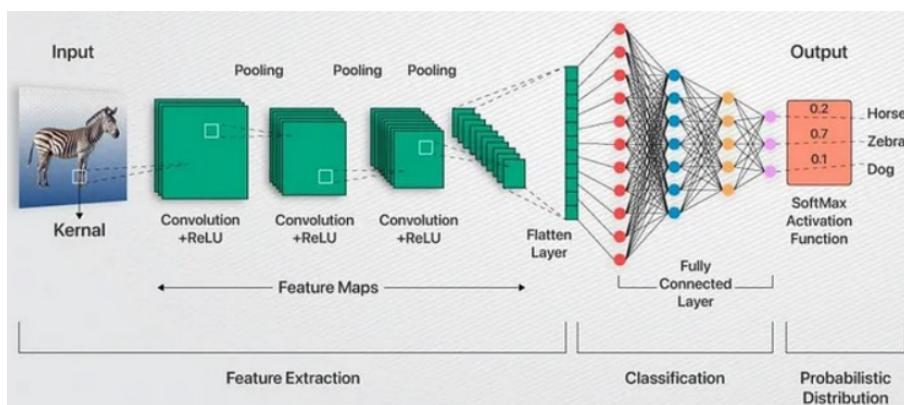
Pored slojeva sa gustim vezama, koji kombinuju sve ulaze sa svim neuronima u sledećem sloju, neuronske mreže koriste i druge specijalizovane tipove slojeva, prilagođene različitim vrstama podataka i zadacima.

Konvolucijski slojevi (*Convolutional Layers*) koriste filtere ili jezgra (*kernels*) za transformaciju podataka, posebno kada se radi o slikama ili video sekvencama. Svaki filter uči karakteristične obrasce u podacima, a vrednosti filtera se ažuriraju tokom procesa učenja, slično težinama i odstupanjima u gustim slojevima. Ovi slojevi su standard u zadacima računarskog vida (*computer vision*), jer omogućavaju automatsko

prepoznavanje ivica, tekstura i drugih vizuelnih osobina koje služe za klasifikaciju ili detekciju objekata.



Slika 10.5: Rekurentna neuronska mreža (RNN): prenos informacija duž sekvence podataka.



Slika 10.6: Konvolucijski sloj: primena filtera (jezgra) nad slikom radi ekstrakcije karakteristika.

Rekurzivni slojevi (*Recurrent Layers*) uče o odnosima između elemenata u nizu podataka. Neuron u ovim slojevima prenose naučene informacije duž sekvence, što omogućava modelu da povezuje prethodne ulaze sa kasnijim izlazima. Na primer, za ulazni niz X_0, X_1, X_2 odgovarajući izlazi Y_0, Y_1, Y_2 se generišu tako da neuron koji procesira X_1 uzima u obzir i prethodno stanje neurona koji je procesirao X_0 , čime se prenosi kontekst. Ova osobina omogućava mrežama predviđanje sekvenci ili analizu vremenskih serija.

Jedan od popularnih tipova rekurzivnih slojeva je LSTM (*Long Short-Term Memory*), koji ne samo da prenosi vrednost prethodnog neurona na sledeći, već čuva informacije sa udaljenijih ulaza u sekvenci. Na primer, ulaz X_0 može uticati na izlaz Y_9 , što omogućava modelu da uči dugoročne zavisnosti u podacima. Takvi slojevi su posebno korisni za predviđanje vremenskih serija, obradu jezika i razumevanje teksta.

Postoje i slojevi koji ne uče parametre, ali utiču na tok podataka i efikasnost mreže. Slojevi poput *dropout* smanjuju gustinu veza između neurona tokom učenja kako bi se sprečilo preprilagođavanje (*overfitting*). Slojevi za udruživanje (*pooling*) redukuju dimenzije podataka, zadržavajući ključne informacije i smanjujući redundanciju. Lambda slojevi omogućavaju izvršavanje proizvoljnih transformacija nad podacima unutar mreže.

Primenjujući princip učenja koji je ilustrisan na jednostavnom primeru jednog neurona sa formulom $Y = wX + b$, ista metodologija se koristi i u složenijim mrežama. Težine i odstupanja u svakom neuronu se iterativno ažuriraju prema funkciji gubitka, dok višeslojna arhitektura omogućava modeliranje nelinearnih i složenih odnosa između ulaza i izlaza. Na ovaj način, fundamentalni principi učenja iz prostih linearnih modela direktno se primenjuju i u naprednijim neuronskim mrežama sa konvolucijskim i rekurzivnim slojevima.

10.5 Osnovni blokovi neuronskih mreža: inicijalizacija i učenje

Učenje u mašinskom učenju podrazumeva početni skup podataka i pripadajuće izlaze. Zadaci algoritma sastoji se u pronalaženju pravila koja povezuju ulaze i izlaze kako bi model mogao da primeni naučene obrasce na novim podacima.

Jednostavan primer ovog principa može se predstaviti linearnom funkcijom $Y = wX + b$, gde w predstavlja težinu koja modifikuje ulaznu vrednost, a b označava odstupanje (*bias*). Početni podaci mogu biti, na primer, niz brojeva $[-1, 0, 1, 2, 3, 4]$ sa odgovarajućim izlazima $[-3, -1, 1, 3, 5, 7]$. U ovoj fazi, vrednosti w i b se mogu inicijalno odabrati proizvoljno, a funkcija gubitka izračunava se kao srednja kvadratna greška između predviđenih i stvarnih vrednosti. Cilj procesa učenja je da se ova funkcija minimizira, čime se određuju optimalne vrednosti parametara.

Minimizacija funkcije gubitka odvija se korišćenjem gradijenata. Izvod funkcije gubitka u odnosu na parametre w i b pokazuje smer kretanja ka minimumu. Veličina koraka u tom smeru definiše se kao brzina učenja (*learning rate*). Pomoću trenutne lokacije na funkciji i vrednosti brzine učenja određuju se nove vrednosti parametara, koje se zatim koriste u sledećem krugu procene greške. Ovaj iterativni proces naziva se propagacija unazad (*backpropagation*), dok se kretanje niz funkciju gubitka ka njenom minimumu naziva gradijentni pad (*gradient descent*).

Kod višeslojnih neuronskih mreža, pojedinačni neuroni se organizuju u slojeve, pri čemu svaki neuron ima sopstvene težine i odstupanje. Izlaz neurona sa više ulaza može se izraziti formulom $Y = \sum_i w_i k_i + b$, gde su k_i izlazi neurona iz prethodnog sloja, a w_i njihove težine. Naučene vrednosti težina i odstupanja određuju se tokom procesa učenja, omogućavajući mreži da povezuje ulaze i izlaze sa minimalnom greškom.

Iako se u praktičnim implementacijama koriste softverski okviri koji automatski obavljaju izračunavanje težina, kompilaciju modela i definisanje funkcije gubitka, suštinski princip ostaje isti kao kod linearne funkcije. Čak i u složenijim mrežama, metodologija inicijalizacije parametara, izračunavanja greške, propagacije unazad i gradijentnog pada primenjuje se na identičan način.

Ovakav pristup omogućava neuronskim mrežama da, kombinovanjem više neurona i slojeva, uče složene funkcije i obrasce. Dok u jednostavnoj linearnoj mreži jedan neuron odgovara funkciji sa dva parametra, u višeslojnim mrežama set težina i odstupanja u slojevima omogućava mreži da rešava zadatke znatno kompleksnije od samog linearno definisanog skupa podataka.

Glava 11

Neuronske mreže

Cilj poglavlja

Cilj ovog poglavlja je da se pruži detaljno objašnjenje principa funkcionisanja neuronskih mreža, kao i načina na koji se ostvaruje proces učenja i prilagođavanja parametara modela. U okviru poglavlja razmatra se inicijalizacija i obuka modela na primeru linearne regresije, čime se uvodi osnovni koncept optimizacije pomoću gradijentnog pada i propagacije unazad. Posebna pažnja posvećena je proširenju ovog principa na mreže sa više neurona i slojeva, koje omogućavaju formiranje složenijih i nelinearnih funkcija.

Dodatno se obrađuju osnove klasifikacije i uloga neuronskih mreža u prepoznavanju obrazaca, uz primenu na skupu podataka MNIST. Objasnjava se struktura i proces obuke klasifikacionih mreža, kao i podela podataka na skupove za učenje, validaciju i testiranje. Na kraju se uvode principi i funkcionalnost konvolucionih neuronskih mreža, sa naglaskom na obradu slikovnih podataka, ekstrakciju karakteristika i smanjenje dimenzionalnosti korišćenjem konvolucionih i *pooling* slojeva.

11.1 Inicijalizacija i učenje na primeru linearne regresije

Mašinsko učenje predstavlja proces u kojem su na početku poznati podaci i njihovi odgovarajući odgovori, dok je zadatak algoritma da

otkrije pravilo koje ih povezuje, kako bi to pravilo kasnije moglo da se primeni na nove, nepoznate podatke.

Jedan ilustrativan primer ovog principa može se prikazati ako se pretpostavi da su dostupni podaci u vidu skupa vrednosti

$$X = [-2, -1, 0, 1, 2, 3]$$

i odgovarajući skup izlaznih vrednosti

$$Y = [-5, -3, -1, 1, 3, 5].$$

Ako se pretpostavi da je odnos između ulaznih i izlaznih vrednosti linearan, on se može izraziti funkcijom:

$$Y = wX + b, \tag{11.1}$$

gde w označava težinski koeficijent, a b slobodni član. Cilj procesa učenja jeste određivanje optimalnih vrednosti parametara w i b .

Postupak određivanja rešenja može da započne proizvoljnim izborom početnih vrednosti za w i b , nakon čega se procenjuje greška, odnosno *funkcija gubitka* (loss), koja meri odstupanje između tačnih i predviđenih vrednosti. Na primer, ako su $w = 8$ i $b = -4$, dobijaju se predviđene vrednosti:

$$\hat{Y} = [-20, -12, -4, 4, 12, 20],$$

dok su stvarne vrednosti:

$$Y = [-5, -3, -1, 1, 3, 5].$$

Za svaku tačku odstupanje se može izraziti kao:

$$e_i = \hat{y}_i - y_i. \tag{11.2}$$

Ukupna greška može da se aproksimira kao srednja vrednost svih odstupanja. Međutim, pošto neka odstupanja imaju pozitivne, a neka negativne vrednosti, njihovo prostim sabiranjem moglo bi doći do međusobnog poništavanja. Zbog toga se u praksi koristi *srednja kva-*

dratna greška (engl. *Mean Squared Error*, MSE), definisana kao:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2, \quad (11.3)$$

gde N označava ukupan broj uzoraka. Ova funkcija gubitka ima oblik parabole, što omogućava njenu optimizaciju.

Cilj procesa učenja jeste minimizacija funkcije gubitka. Smer kretanja ka minimumu određuje se izračunavanjem gradijenta funkcije gubitka u odnosu na parametre w i b :

$$\nabla \mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial w}, \frac{\partial \mathcal{L}}{\partial b} \right). \quad (11.4)$$

Veličina koraka kojim se prilazi minimumu određuje se parametrom *learning rate*, koji se obično označava kao η .

Na osnovu trenutne pozicije na funkciji gubitka i veličine koraka, izračunavaju se nove vrednosti parametara w i b :

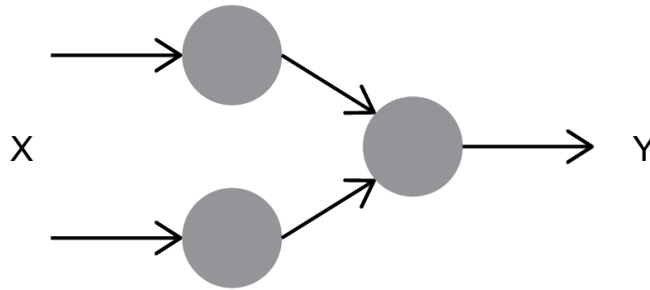
$$\begin{aligned} w_{\text{novo}} &= w_{\text{staro}} - \eta \frac{\partial \mathcal{L}}{\partial w}, \\ b_{\text{novo}} &= b_{\text{staro}} - \eta \frac{\partial \mathcal{L}}{\partial b}. \end{aligned} \quad (11.5)$$

Ovaj postupak ažuriranja parametara poznat je kao *propagacija unazad* (engl. *backpropagation*). Proces postupnog približavanja minimumu funkcije gubitka korišćenjem gradijenta naziva se *gradijentni pad* (engl. *gradient descent*).

11.2 Funkcija $f(x)$ sa više od jednog neurona

Umesto korišćenja samo jednog neurona i jednostavne linearne funkcije, veza između ulaza x i izlaza y može se ostvariti pomoću *neuronske mreže* (NN) koja sadrži više neurona. Na slici je prikazan jednostavan primer takve mreže u kojoj prvi sloj sadrži dva neurona, dok drugi sloj ima jedan neuron.

U softverskom okviru TensorFlow/Keras ovakva mreža može se opisati sledeće:



Slika 11.1: Jednostavna neuronska mreža sa dva neurona u skrivenom sloju i jednim izlaznim neuronom.

```
model = tf.keras.Sequential([
    keras.layers.Dense(2, input_shape=[1]),
    keras.layers.Dense(1)
])
```

Model se zatim kompajlira i obučava korišćenjem standardnih funkcija optimizacije i gubitka, kao i u slučaju linearne regresije.

Neuron u drugom sloju ima dva ulaza, pa se njegov izlaz može izraziti kao:

$$y = w_1 k_1 + w_2 k_2 + b, \quad (11.6)$$

gde su k_1 i k_2 izlazi neurona iz prethodnog sloja, dok su w_1 i w_2 njihove odgovarajuće težine. Parametar b predstavlja slobodni član (bias). Ako se u prethodnom sloju nalazi više neurona, broj težina raste proporcionalno broju ulaza.

Nakon učenja, vrednosti težina i slobodnih članova mogu se ispitati pozivom funkcije `get_weights()`, koja vraća naučene parametre slojeva. Na primer, za prvi sloj mogu se dobiti dve vrednosti težina i dve vrednosti slobodnih članova, dok drugi sloj sadrži onoliko težina koliko ima neurona u prethodnom sloju, uz jedan bias.

Ove težine i bias sabiraju se prema izrazu (11.6), čime se formira izlaz neurona drugog sloja:

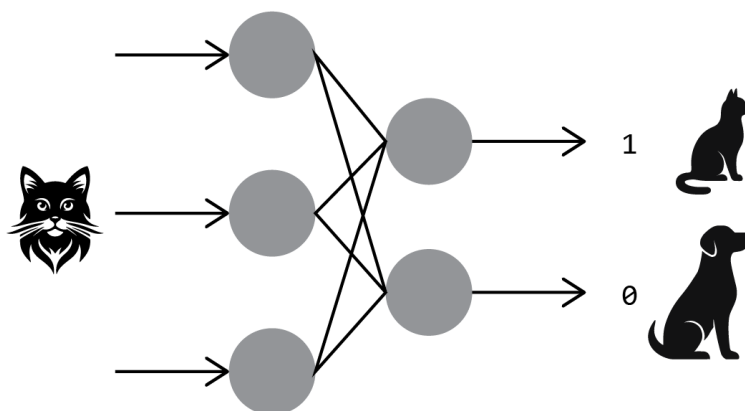
$$y = w_1 k_1 + w_2 k_2 + b. \quad (11.7)$$

Na ovaj način neuronska mreža sa više neurona u slojevima omo-

gućava formiranje složenijih, potencijalno nelinearnih funkcija, čime se znatno povećava sposobnost modela da aproksimira različite odnose između ulaza i izlaza.

11.3 Uvod u klasifikaciju

U prethodnim primerima razmatrana je regresija, gde je cilj bio predviđanje numeričke vrednosti na osnovu jednog ili više ulaznih parametara. Druga uobičajena primena neuronskih mreža jeste *klasifikacija*, u kojoj se mreža koristi za određivanje klase kojoj pripada dati podatak.



Slika 11.2: Primer binarne klasifikacije: neuronska mreža prima sliku i odlučuje da li prikazuje psa ili mačku.

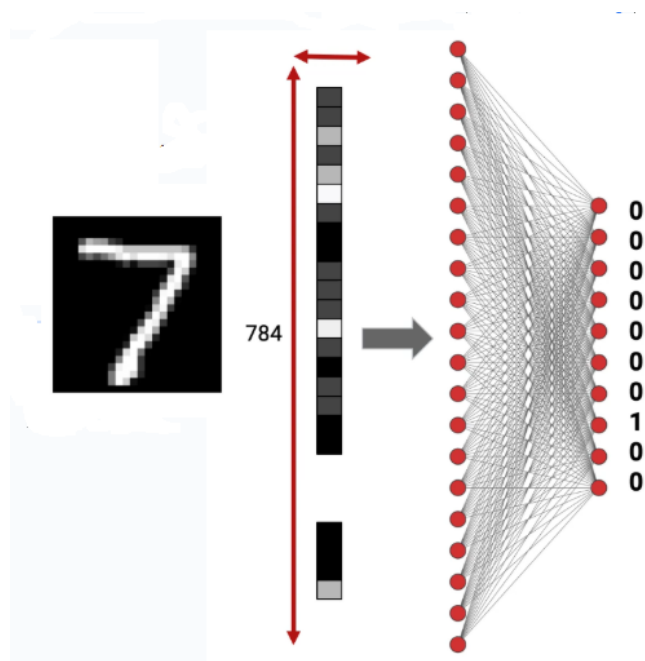
Za razliku od regresionih mreža koje imaju jedan izlazni neuron, u klasifikaciji se obično koristi više neurona u izlaznom sloju, pri čemu svaki neuron odgovara jednoj klasi. Na primer, za zadatak razlikovanja slika mačaka i pasa, izlazni sloj može sadržati dva neurona — jedan za klasu „mačka“ i jedan za klasu „pas“. Ako mreži prosledimo sliku mačke, neuron za mačku će imati visoku verovatnoću aktivacije, dok će neuron za psa biti slabije aktiviran. Oznake za obuku mogu biti u formi vektora, npr. $[1, 0]$ za mačku i $[0, 1]$ za psa.

Sličan princip koristi se i za mreže koje prepoznaju rukom napisane cifre od 0 do 9. Za ovakav zadatak izlazni sloj mreže sadrži deset

neurona, po jedan za svaku cifru.

11.3.1 Skup podataka MNIST

Za učenje se koristi poznati skup podataka *MNIST*, koji sadrži 60 000 slika za obuku i 10 000 slika za validaciju. Svaka slika je crno-bela i dimenzija 28×28 piksela. Radi lakšeg učenja, vrednosti piksela normalizuju se tako da budu u opsegu $[0, 1]$.



Slika 11.3: Primer MNIST slike: broj 7 i njene klasifikacije

11.3.2 Struktura mreže

Neuronska mreža za prepoznavanje cifara može se predstaviti kao složena, ali logički jednostavna struktura:

- Prvi sloj, *Flatten*, pretvara dvodimenzionalnu sliku u jednodimenzionalni vektor od 784 elemenata, što omogućava njegovo dalje procesiranje u gustim slojevima.

- Sledi gusti sloj sa 20 neurona koji koristi aktivacionu funkciju *ReLU* (Rectified Linear Unit), čime se uvodi nelinearnost i sprečava međusobno poništavanje vrednosti neurona.
- Izlazni sloj ima 10 neurona sa aktivacionom funkcijom *softmax*, koja pretvara izlaze u verovatnoće, a neuron sa najvećom vrednošću označava predviđenu cifru.

Ova struktura omogućava mreži da nauči obrasce na slikama i da klasifikuje cifre prema naučenim karakteristikama.

11.3.3 Obuka i evaluacija

Mreža se obučava korišćenjem optimizatora *Adam* i funkcije gubitka *sparse categorical crossentropy*, koja je pogodna za zadatke sa više kategorija. Proces obuke uključuje prolazak kroz sve slike u skupu podataka (epohe), pri čemu mreža iterativno prilagođava težine svojih neurona kako bi minimizovala grešku klasifikacije.

Nakon obuke, mreža se testira na validacionom skupu. Rezultat je vektor verovatnoća za svaku sliku, pri čemu najveća vrednost pokazuje koju cifru mreža prepoznaje. Na primer, za sliku cifre 7, neuron odgovarajući cifri 7 može imati verovatnoću blizu 99%, dok ostali neuroni imaju znatno niže vrednosti. Ovo potvrđuje da mreža uspešno klasifikuje slike na osnovu naučenih obrazaca.

11.4 Podaci za učenje, validaciju i testiranje

Prilikom rada sa neuronskim mrežama dostupni podaci se dele u tri osnovne grupe: podaci za učenje, podaci za validaciju i podaci za testiranje.

Podaci za učenje predstavljaju najveći deo skupa podataka i koriste se za prilagođavanje težina mreže tokom procesa obuke. Podaci za validaciju omogućavaju procenu performansi mreže u toku obuke, nakon svakog koraka, i pomažu u podešavanju hiperparametara, kao što su broj slojeva ili veličina koraka učenja. Podaci za testiranje služe za konačnu proveru tačnosti mreže nakon završetka procesa obuke i daju objektivan uvid u njenu sposobnost da generalizuje na nove, nepoznate podatke.

Važno je napomenuti da postizanje vrlo visoke tačnosti na podacima za učenje, na primer 99%, ne garantuje dobru generalizaciju. U takvim slučajevima često dolazi do prevelike zavisnosti modela od konkretnih podataka za učenje, što može rezultovati značajno nižim vrednostima tačnosti na skupovima za validaciju i testiranje.

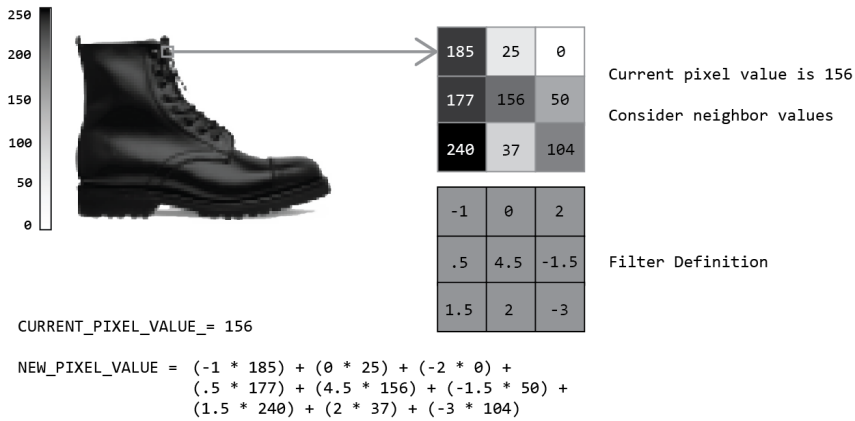
11.5 Konvolucione neuronske mreže

Konvolucione neuronske mreže (*Convolutional Neural Networks*, *CNN*) koriste se posebno u oblasti *computer vision*, gde *dense layers* i *flatten layers* nisu dovoljni za efikasno prepoznavanje obrazaca na slikama. Osnovni gradivni element *CNN*-a su *convolutional layers*, koji omogućavaju automatsko izdvajanje karakterističnih osobina slike (*feature extraction*). Pored standardnih parametara *weights* i *bias*, u okviru *CNN*-a uče se i vrednosti *filters* koji se primenjuju pri konvoluciji.

Filtri funkcionišu tako što se za svaki piksel slike uzimaju i njegovi neposredni susedi. Svaki piksel i njegov sused se množe sa odgovarajućom vrednošću iz *filtera*, a rezultati se zatim sumiraju. Dobijena vrednost postaje nova vrednost piksela u izlaznoj slici. Na ovaj način filteri mogu da istaknu određene karakteristike slike, kao što su vertikalne ili horizontalne linije, ivice i oblici. Sličan princip se koristi i u programima za obradu slike, kao što su Photoshop ili Instagram, za primenu vizuelnih filtera.

Proces izdvajanja značajnih osobina slike naziva se *feature extraction*. Kada se ekstrakcija kombinuje sa operacijom *pooling*, mreža može smanjiti dimenzije slike, zadržavajući najvažnije informacije. Najčešće se koristi *max pooling*, gde se iz malih blokova piksela bira najveća vrednost, dok se ostale odbacuju. Na primer, blok dimenzija četiri puta četiri piksela može se podeliti na četiri manja bloka od po dva puta dva piksela, a iz svakog se bira maksimum. Rezultat je novi, komprimovani prikaz slike sa očuvanim karakteristikama. Objedinjavanje se može izvoditi i pomoću minimuma (*min pooling*) ili proseka (*average pooling*), u zavisnosti od potrebe.

Primena više *filters* omogućava mreži da stvori različite varijante slike i više informacija za učenje. Na primer, filteri mogu izdvojiti linije, oblike ruku, nogu i lica, što mreži omogućava da prepozna prisustvo osobe na slici. Vrednosti filtera se inicijalno postavljaju nasumično, a tokom obuke se prilagođavaju tako da mreža uči da izdvoji karakteri-



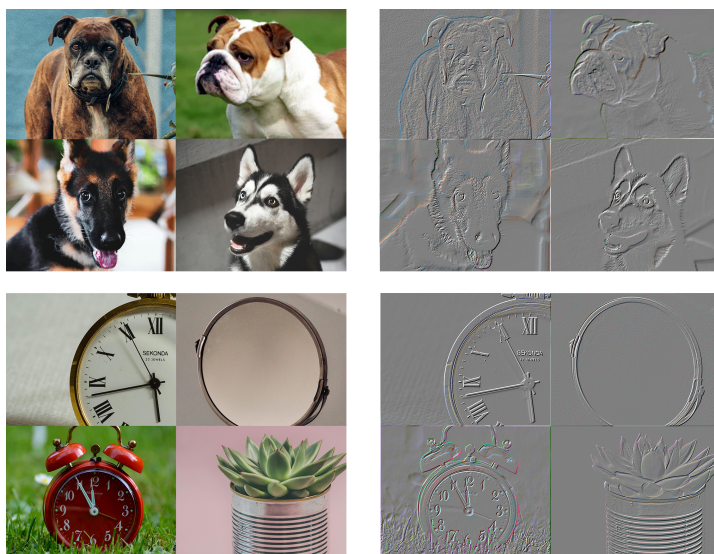
Slika 11.4: Ilustracija primene filtera u CNN-u. Filter (kernel) klizi po ulaznoj slici, množi se sa vrednostima piksela i sumira da bi se dobila izlazna mapa karakteristika (*feature map*). Više filtera omogućava detekciju različitih obrazaca.

stike koje su relevantne za dodeljivanje klase (*class labels*).

Računanje sa *filters* i sa *pooling*-om je ključno za funkcionalnost CNN-a. Korišćenjem *filters* mreža identifikuje karakteristične obrasce, dok *pooling* omogućava smanjenje dimenzija i jačanje značajnih karakteristika, čime se postiže efikasnija obrada i smanjuje složenost podataka.

Kao i kod standardnih *weights* i *bias*, *filters* se podešavaju korišćenjem *optimizer*-a. Kada mreža nauči da određeni filteri izdvajaju karakteristike poput nogu, ruku i glave, oni postaju povezani sa klasom osoba. Na ovaj način mreža može prepoznati slike i u situacijama kada karakteristike nisu očigledne ljudskom oku. Neki filteri otkrivaju obrasce piksela koji ljudima ne znače ništa, ali su konzistentni i korisni za model.

Detaljnije objašnjenje kako *CNN* vidi karakteristike slike može se naći u radu <https://arxiv.org/pdf/1311.2901v3.pdf>. U ovom radu su prikazani rezultati primene filtera na slike pasa i zakrivljenih predmeta, kao i rekonstrukcije slika generisanih mrežom vraćanjem kroz *filters*. Primeri koda za vizuelizaciju filtera dostupni su na https://keras.io/examples/vision/visualizing_what_convnets_learn/.



Slika 11.5: Detekcija ivica pomoću CNN.

Glava 12

Primene Edge AI i TinyML

Cilj poglavlja

Cilj ovog poglavlja jeste prikaz koncepta Edge AI i TinyML tehnologija, kao i analiza njihovih mogućnosti primene u industrijskim, zdravstvenim i ekološkim sistemima. Poseban akcenat stavljen je na ulogu senzora u procesima prikupljanja i lokalne obrade podataka, kao i na značaj odgovornog i etičkog dizajna u razvoju inteligentnih uređaja ograničenih resursa.

12.1 Uvod

Embedded mašinsko učenje i njegove podvrste Edge AI i TinyML omogućavaju obradu podataka direktno na uređaju, umesto slanja u cloud, što je posebno korisno za uređaje sa ograničenim resursima za obradu i povezivanje. Prednosti ove paradigme uključuju prenosivost uređaja, male vrednosti kašnjenja, smanjenu potrebu za propusnim opsegom za prenos, privatnost i niže troškove.

U realnom svetu, Edge AI se primenjuje u različitim oblastima, od industrije i zdravstva do ekologije i svakodnevnih uređaja. Tipični primeri uključuju detekciju kvarova na dalekovodima radi prevencije šumskih požara, zaštitu spasilaca putem inteligentnih nosivih uređaja ili praćenje ponašanja životinja pomoću pametnih ogrlica.



Slika 12.1: Edge AI koncept – uređaji sa senzorima i lokalnom obradom podataka

12.2 Uloga senzora u TinyML

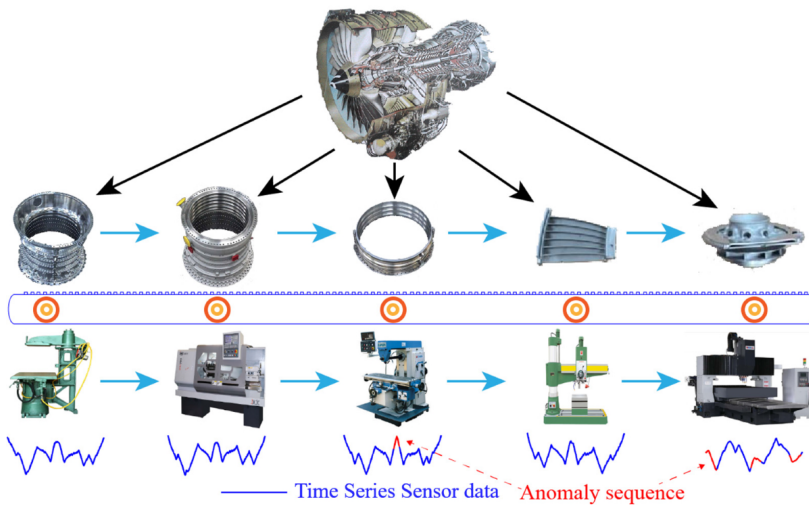
Senzori predstavljaju ključnu vezu između algoritamskih modela i fizičkog sveta. Podaci sa senzora mogu biti vremenske serije, slike ili kombinacije različitih signala. Najčešće korišćeni senzori u TinyML aplikacijama su akustični senzori, vizuelni senzori, akcelerometri, žiroskopi, temperaturni i vibracioni senzori, kao i biosenzori za praćenje ljudskog ili životinjskog zdravlja.

Kombinovanjem više senzora može se dobiti veća preciznost i dodatna analiza složenih pojava. Na primer, akcelerometar i žiroskop u mobilnom uređaju omogućavaju detekciju šestosmernih pokreta, dok kombinacija kamere i mikrofona može pratiti ljudske aktivnosti ili ponašanje životinja. Senzori imaju ograničen radni vek, mogu pokazivati šum i drift, a previše senzora povećava potrošnju energije i složenost sistema.

12.3 Industrijske primene

U industrijskim okruženjima, Edge AI i TinyML omogućavaju praćenje i kontrolu sistema u realnom vremenu, povećavajući otpornost i pouzdanost. Prediktivno održavanje koristi senzore vibracija, akcelerometre, temperature i audio signale kako bi se identifikovale anomalije u

radu mašina pre nego što dođe do kvara.



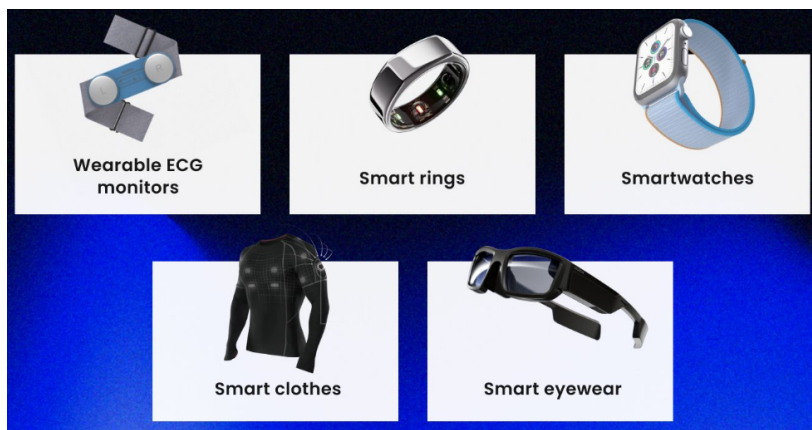
Slika 12.2: Mašine opremljene senzorima za prediktivno održavanje

Autonomna kontrola alata i vozila oslanja se na vizuelne i pokretne senzore, omogućavajući robotima i autonomnim kombajnima da obavljaju zadatke bez stalnog nadzora. Detekcija neispravnih proizvoda na proizvodnim linijama koristi vizuelne senzore, dok analiza pokreta u skladištima kombinuje više senzora radi optimizacije logistike. TinyML modeli se treniraju da prepoznaju obrasce u podacima sa senzora i donose odluke bez potrebe za cloud konekcijom.

12.4 Primene u zdravstvu i ljudskim aktivnostima

Nosivi uređaji prate ljudsko zdravlje ili fizičku aktivnost u realnom vremenu. Podaci sa akcelerometara, GPS-a, biosenzora i mikrofona omogućavaju praćenje spavanja, detekciju padova, kategorizaciju fizičkih aktivnosti i prepoznavanje gestova.

Uređaji mogu obavestiti medicinsko osoblje ili korisnika o anomalijama bez potrebe za internet konekcijom, čime se povećava privatnost i sigurnost podataka.



Slika 12.3: Nosivi uređaji prate fizičke aktivnosti i biosignale korisnika

12.5 Primene u ekologiji i životinjskom svetu

Praćenje životinja i prirodnog okruženja koristi TinyML za kontinuirano i lokalno prikupljanje podataka. Pametne ogrlice i vizuelni senzori omogućavaju praćenje zdravlja domaćih životinja i divljih životinja, dok termalne kamere i vizuelni senzori omogućavaju detekciju divljih životinja ili potencijalnih opasnosti.

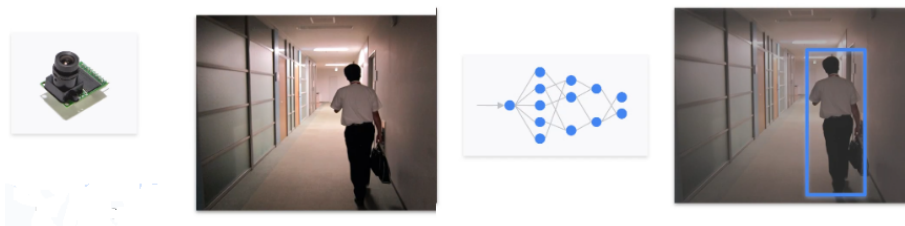


Slika 12.4: Pametna ogrlica za životinje koja prati aktivnost i zdravlje

Takođe, vizuelni senzori se koriste za dijagnostiku bolesti useva ili monitoring poljoprivrednih resursa. Kombinacija više senzora omogućava modelima da prepoznaju obrasce koji ljudima možda nisu očigledni, ali su značajni za klasifikaciju i donošenje odluka.

12.6 Obrada audio i vizuelnih signala

Edge AI i TinyML omogućavaju lokalnu obradu audio, vizuelnih i vremenskih senzorskih signala. Audio signali se koriste za detekciju ključnih reči, filtriranje buke, generisanje govora i muzike, dok vizuelni signali omogućavaju detekciju objekata, segmentaciju pozadine i kompresiju video sadržaja.



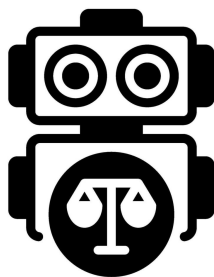
Slika 12.5: Obrada vizuelnih signala na uređaju

Lokalna obrada smanjuje latenciju, minimalizuje potrebu za propusnim opsegom i omogućava kontinuirani rad uređaja na baterijsko napajanje.

12.7 Dizajn i odgovornost

Razvoj aplikacija u okviru Edge AI i TinyML mora biti odgovoran i etički. Modeli moraju biti nepristrasni, pouzdani i bezbedni, naročito u kritičnim oblastima kao što su autonomna vozila, industrijska automatizacija i medicinski uređaji. Tokom čitavog životnog ciklusa razvoja – od prikupljanja podataka, izbora senzora i treniranja modela, do implementacije i korišćenja – potrebno je obezbediti transparentnost, pravičnost i zaštitu privatnosti. Odgovoran AI dizajn podrazumeva razmatranje tri ključne faze razvoja TinyML sistema: dizajna, razvoja i upotrebe. U fazi dizajna postavljaju se pitanja šta se kreira, za koga

i sa kojim posledicama ako sistem ne funkcioniše pravilno. U tom kontekstu, AI rešenja bi trebalo da teže ostvarivanju društvenog dobra i da budu u skladu sa ciljevima održivog razvoja Ujedinjenih nacija, kao što su smanjenje nejednakosti, unapređenje zdravlja, obrazovanja i zaštite životne sredine. Posebnu pažnju potrebno je posvetiti potencijalnim zloupotrebama i etičkim rizicima, uključujući narušavanje privatnosti i neovlašćeno profilisanje korisnika. Tokom razvoja treba razmotriti ograničenja podataka, pristrasnost u treniranju modela i ograničene resurse ugrađenih uređaja, dok faza korišćenja obuhvata procenu grešaka sistema i njihove posledice. Tip 1 (lažno pozitivne) i Tip 2 (lažno negativne) greške različito utiču na bezbednost i pouzdanost, pa je u dizajnu neophodno proceniti koja od njih može imati ozbiljnije posledice za konkretnu primenu, poput detekcije požara, medicinskih alarma ili sistema prepoznavanja govora. Samo uz ovakav sveobuhvatan pristup moguće je razvijati TinyML rešenja koja su istovremeno efikasna, pouzdana i društveno odgovorna.



Slika 12.6: Simbol odgovornog i etičkog dizajna AI sistema

Literatura

- [1] M. Kranz, *Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry*. Hoboken, NJ: Wiley, 2016.
- [2] NXP Semiconductors, *I2C-bus specification and user manual*, 2014. UM10204 – publicly available from NXP website.
- [3] MIPI Alliance, “Mipi i3c basic specification,” 2019. MIPI I3C Basic (publicly available summary/specs on MIPI site).
- [4] Motorola industry, “Serial peripheral interface (spi) overview.” SPI descriptions and application notes.
- [5] R. Heydon, *Bluetooth Low Energy: The Developer’s Handbook*. Portland, OR: Pragmatic Bookshelf, 2012.
- [6] LoRa Alliance, “Lorawan specification and resources.” LoRaWAN overview and specs available on LoRa Alliance website.
- [7] J. Yiu, *The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors*. Oxford, UK: Newnes (Elsevier), 2 ed., 2013.
- [8] J. W. Valvano, *Embedded Systems: Introduction to ARM Cortex-M Microcontrollers*. USA: CreateSpace Independent Publishing Platform, 2017.
- [9] D. Harris and S. Harris, *Digital Design and Computer Architecture*. San Francisco, USA: Morgan Kaufmann, 2 ed., 2012.
- [10] Najm, D. Harris, and A. Weste, *CMOS VLSI Design: A Circuits and Systems Perspective*. Boston, USA: Addison-Wesley, 4 ed., 2010.

- [11] P. Horowitz and W. Hill, *The Art of Electronics*. Cambridge, UK: Cambridge University Press, 3 ed., 2015.
- [12] P. Warden and D. Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Embedded Devices*. Sebastopol, CA: O'Reilly Media, 2019.
- [13] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. Sebastopol, CA: O'Reilly Media, 2 ed., 2019.
- [14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.
- [15] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, *Efficient Processing of Deep Neural Networks*. San Rafael, CA: Morgan Claypool Publishers, 2020.
- [16] C. Huyen, *Designing Machine Learning Systems*. Sebastopol, CA: O'Reilly Media, 2020.
- [17] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.
- [18] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Chao, and et al., “In-datacenter performance analysis of a tensor processing unit,” in *Proceedings of the 44th International Symposium on Computer Architecture (ISCA)*, pp. 1–12, ACM, 2017.
- [19] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Boston, MA: Addison-Wesley, 2010.
- [20] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*. San Francisco, USA: Morgan Kaufmann, 6 ed., 2017.
- [21] R. E. Bryant and D. R. O'Hallaron, *Computer Systems: A Programmer's Perspective*. Boston, MA: Pearson, 3 ed., 2015.
- [22] J. W. S. Liu, *Real-Time Systems*. Upper Saddle River, NJ: Prentice Hall, 2000.

- [23] Real Time Engineers Ltd., “Freertos – real time kernel for micro-controllers,” 2024.
- [24] Arm Mbed, “Mbed os – an open-source rtos for embedded devices,” 2024.
- [25] ARM Ltd., *ARM Architecture Reference Manual*. ARM architecture documents and technical reference manuals (vendor site).
- [26] TensorFlow / Google, “Tensorflow lite and tensorflow lite for microcontrollers documentation,” 2024.

Ključne reči

Ovo poglavlje daje proširene definicije ključnih termina koji se koriste u razmatranjima ugrađenog hardvera, komunikacionih sistema i primene mašinskog učenja u embedded uređajima. Služi kao leksikon radi brze reference pri čitanju teksta.

Elektronske komponente i senzori

Otpornik (resistor): pasivna komponenta koja ograničava električni tok u kolu. Vrednost u omima, parametri: tolerancija, snaga, temperaturni koeficijent.

Kondenzator (capacitor): skladišti električni naboj; koristi se za filtriranje, stabilizaciju napona i formiranje vremenskih konstanti.

Dioda (diode): provodi struju primarno u jednom smeru; specijalne vrste: LED, Schottky, Zener.

Tranzistor (transistor): pojačavač ili prekidač; BJT i MOSFET. Osnovni gradivni blok digitalne i analogne elektronike.

Operacioni pojačavač (op-amp): integrisano pojačalo za pojačavanje, filtriranje, integraciju i diferenciranje signala.

Akcelerometar, žiroskop, magnetometar, mikrofonski, kamera, senzori gasa, IR senzori, kapacitivni senzori: prate okolinu i omogućavaju interakciju uređaja sa svetom.

Napajanje i upravljanje energijom

Baterije i akumulatori: izvori DC napona (Li-ion, NiMH) sa različitim energijskom gustinom i ciklusnim trajanjem.

Regulator napona i DC-DC konvertori (buck/boost): održavaju stabilan napon za rad uređaja.

Sakupljanje energije (energy harvesting): solar, piezo, termoelektrični generatori, RF harvesting za napajanje niskopotrošnih uređaja.

Programabilni i specijalizovani procesori

Mikrokontroler (MCU): CPU, memorija i periferni kontroleri na jednom čipu, namenjen upravljanju uređajima.

FPGA: programabilna logička matrica za implementaciju hardverske logike nakon proizvodnje.

ASIC: namenski dizajnirano i trajno IC za specifičnu aplikaciju.

GPU: paralelni procesor optimizovan za vektorske i matriksne operacije; primena u vizuelnim i ML zadacima.

SoC: integracija CPU, memorije i dodatnih blokova (AI akceleratori, komunikacije) na jednom čipu.

VPU, NNP, MLP: specijalne jedinice za ubrzanje računalnog vida, neuronskih mreža i generalnih ML zadataka.

Komunikacioni interfejsi i mreže

UART, SPI, I²C: serijski interfejsi za povezivanje uređaja.

Wi-Fi, Bluetooth/BLE, ZigBee, LoRa/LoRaWAN, NB-IoT: bežične tehnologije za lokalne i široke mreže.

Ethernet, CAN, 2G/3G/4G/5G: žične i mobilne mreže; koristi se u IoT i ugrađenim sistemima.

Mesh, star, bus/topologije: organizacija uređaja u mreži radi optimalne komunikacije.

Edge i cloud arhitekture: kombinacija lokalne obrade i udaljenih serverskih funkcija.

Embedded machine learning i TinyML

TinyML: izvođenje ML modela na uređajima sa ograničenim resursima pri minimalnoj potrošnji energije.

Inferencija (inference): primena već istreniranog modela za predikcije.

Trening (training): offline prilagođavanje parametara modela.

Kvantizacija (quantization): smanjenje preciznosti radi manjih zahteva za memorijom i brže inferencije.

Pruning i kompresija: uklanjanje nepotrebnih težina radi manjih modela i niže potrošnje.

Knowledge distillation: učenje manjeg modela pomoću većeg (teacher → student).

Edge TPU, NPU, akceleratori: ugrađeni hardver za ubrzanje ML zadataka.

Frameworks: TensorFlow Lite, TensorFlow Lite Micro, ONNX, PyTorch Mobile za konverziju i izvođenje modela na embedded platformama.

Alati i softverski koncepti

Debugger, JTAG, SWD: alati za otklanjanje grešaka i praćenje registara.

Bootloader: inicijalno pokretanje i OTA ažuriranja uređaja.

RTOS: deterministički operativni sistem za kritične zadatke.

DMA: direktan prenos podataka bez opterećenja CPU-a.

Power saving modes: režimi niske potrošnje (sleep, deep sleep), DVFS, optimizacija autonomije uređaja.