

## Glava 6

# Arduino i mikrokontroler ATmega328P

U ovom poglavlju ćemo se prvo upoznati sa Arduino ekosistemom, a potom i sa mikrokontrolerom ATmega328P koji se nalazi na platformi Arduino Uno koja se koristi na vežbama iz ovog predmeta.

### 6.1 Arduino

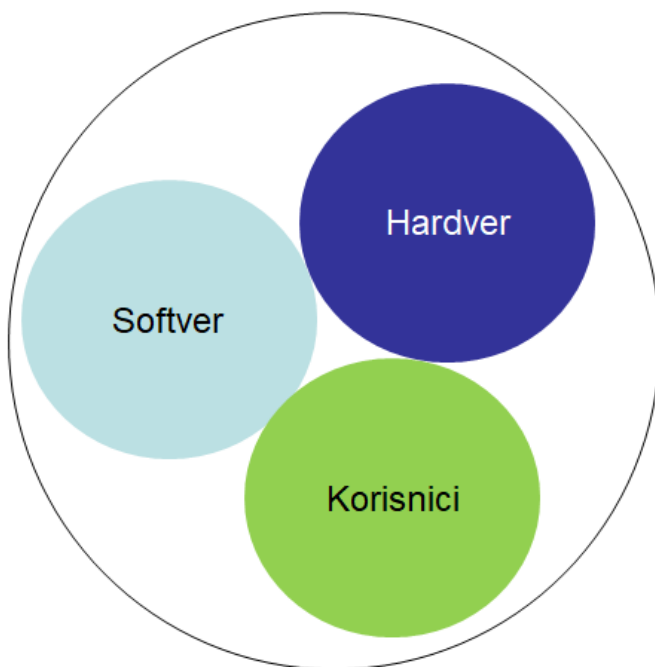
Kada se kaže Arduino obično taj pojam nije jednoznačno određen i mogu se čuti različita objašnjenja (ponekad i netačna). Često se može čuti da je u pitanju neki mikrokontroler ili softver i sl. Šta je Arduino? Uže gledano Arduino predstavlja platformu otvorenog koda (eng. *Open Source*) - razvojni sistem. Na razvojnom sistemu se nalazi odgovarajući mikrokontroler i druge komponente. Šire gledano Arduino predstavlja ekosistem koji se sastoji od odgovarajućeg hardvera, softvera i korisnika (slika 6.1). Može se reći da su ove komponente razlog da je ovakva 'mešavina' veoma uspešna u praksi.

Arduino je nastao kao rezultat saradnje petorice kolega na čelu sa Masimom Banzijem u Italiji. Komercijalni početak se smatra da je bio 2005. godine [31]. Više o tome se može čuti u okviru TED<sup>1</sup> govora Masima Banzija [32]. Broj zvaničnih Arduino ploča na bazi različitih mikrokontrolera (uglavnom AVR Mega) i drugih komponenata trenutno danas broji oko 20tak<sup>2</sup>. Pored zvaničnih postoji jako mnogo

---

<sup>1</sup>konferencije gde autori drže govore pod sloganom 'Ideje vredne širenja'.

<sup>2</sup><https://www.arduino.cc/en/Main/Products>



Slika 6.1: Arduino ekosistem

nezvaničnih realizacija<sup>3</sup>

Neke od karakteristika Arduinoa su:

- Jednostavan je za korišćenje.
- Lako se programira.
- Omogućava brzu izradu prototipova.
- Izgrađen je oko Atmel AVR mikrokontrolera.
- Za rad razvojnog sistema je potreban USB kabel, a kasnije i neke dodatne pločice (tzv. šildovi).

---

<sup>3</sup>Npr. One-chip Arduino <https://www.instructables.com/How-to-Make-a-Single-Chip-Arduino/>

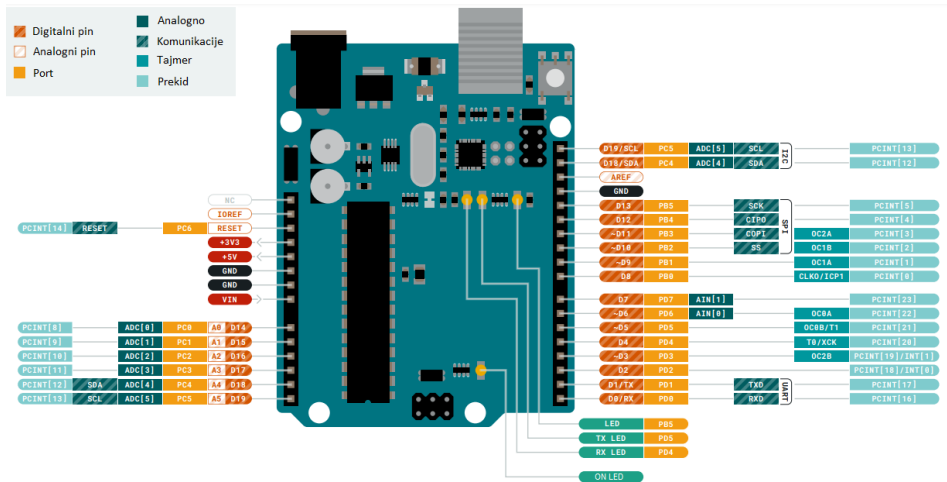


Slika 6.2: Razvojni sistem Arduino Uno

Na vežbama će se koristiti neka od verzija Arduino Uno razvojnog sistema. Izgled ovog razvojnog sistema je prikazan na slici 6.2. Ovo je originalna verzija koja je bazirana na mikrokontroleru ATmega328 koji se nalazi u DIP (eng. *dual in-line package*) sa 28 pinova. Verzije koje nisu originalne češće na sebi imaju verziju mikrokontrolera u TQFP (eng. *thin quad flat pack*) kućištu sa 32 pina. ATmega328P 8-bitni mikrokontroler radi sa učestanošću takta od 16 MHz, ima 2 kB statičke (SRAM) memorije za podatke, 32 kB fleš memorije, 1 kB EEPROM (eng. *Electrically-Erasable/Programmable Read-Only Memory*) memorije, tri LED kojima korisnik može da pristupa, i 20 ulazno/izlaznih pinova. Od toga šest imaju mogućnost da se konfigurišu kao analogni izlazi sa impulsno-širinskom modulacijom (eng. *Pulse-Width Modulation* - PWM), a šest drugih mogu da budu analogni ulazi sa analogno-digitalnom konverzijom. Za povezivanje preko USB porta se na

ploči sa slike 6.2 koristi dodatni mali ATmega16u2 mikrokontroler koji ima unutrašnji USB komunikacioni modul (jer ga ATmega328P nema u sebi). U praksi se sreću i druga rešenja na bazi FTDI232, CH340, CP2102 i dr. čipova. Elektronsku šemu originalne verzije je moguće pogledati na zvaničnoj Arduino stranici [34].

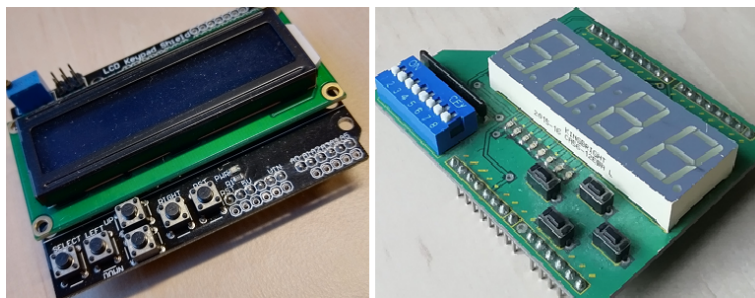
Povezivanje razvojnog sistema sa periferijama se vrši preko konektora koji su za ceo Arduino ekosistem standardizovan. Pinovi na tim konektorima imaju višestruku funkciju zavisno od potreba. Na slici 6.3 je prikazan raspored pinova uz objašnjenja. Osim pinova za napajanje, većina ostalih pinova su u osnovi digitalni pinovi portova B, C i D. Oni pinovi koji ima oznaku tilde ~ ispred, a takvih je 6, imaju mogućnost PWM-a. Šest pinova mogu da se konfigurišu kao analogni ulazi (A0-A5). Nekoliko pinova imaju ulogu pri radu sa tajmerima kao ulaz ili izlaz. Nekoliko pinova se koriste za potrebe različitih tipova serijske komunikacije (I2C, UART, SPI). Svaki pin može da se koristi i pri radu sa prekidima usled promena na pinovima.



Slika 6.3: Arduino Uno raspored pinova

Termin 'šild' (eng. *shield*) predstavlja u Arduino svetu dodatnu štampanu pločicu sa komponentama koja se postavlja na razvojni sistem i omogućava odgovarajuća hardverska proširenja. Najčešće su to dodatne periferije (npr. LCD, sedmo-segmentni displeji, tasteri, prekidači, drajveri motora i sl.). Danas postoji jako mnogo različitih šildova različitih proizvođača. Dobar tutorial o šildovima je napravila kompanija Sparkfun [33].

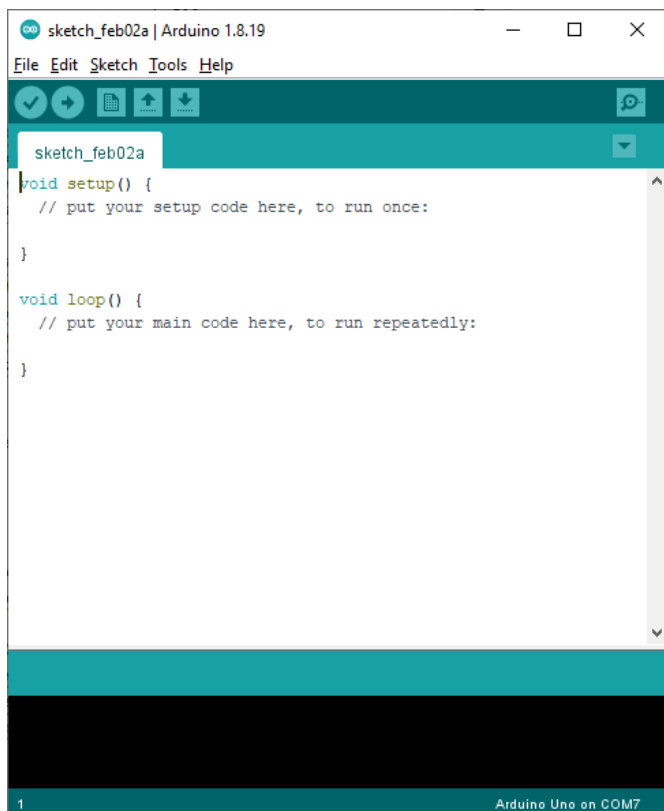
Na vežbama iz ovog predmeta se koriste dva šilda prikazana na slici 6.4. Prvi šild (levi na slici) ima na sebi LCD (eng. *Liquid Crystal Display*) i tastere (6 kom.). Ovaj šild se može kupiti gotov. Drugi šild (na slici desno) sadrži četiri vrste periferija. To su: osam mikroprekidača, osam svetlećih dioda, četiri tastera i četvorocifreni sedmosegmentni displej (sa decimalnim tačkama). Ovaj šild je projektovan na Katedri za elektroniku.



Slika 6.4: Šildovi koji se koriste na vežbama

Za programiranje se koristi Arduino IDE (eng. *integrated development environment*) razvojno okruženje koje je moguće instalirati sa zvanične Arduino stranice [35]. U trenutku pisanja ove knjige je aktuelna verzija 1.8.19. koja se može instalirati na svim glavnim operativnim sistemima. Programiranje se radi u jeziku koji bazira na C++ uz upotrebu različitih biblioteka za rad sa hardverom. Ovo razvojno okruženje podržava jako mnogo različitih razvojnih sistema od kojih su neki odmah podržani prilikom instaliranja, a neke koji nisu iz standardnog Arduino ekosistema je moguće instalirati naknadno. Nakon instaliranja razvojnog okruženja i njegovog aktiviranja pokreće se prozor koji je prikazan sa slici 6.5. Ovde se vidi izgled minimalnog Arduino programa. U Arduino terminologiji se koristi izraz skeč (eng. *sketch*) po analogiji sa slikanjem na slikarskom platnu ili crtanjem u svesci. Minimalni skeč ima jednu proceduru `SETUP()` koja se poziva jednom na početku i služi za inicijalizaciju programskih i hardverskih elemenata. Nakon inicijalizacije se ulazi u beskonačnu petlju `LOOP()` u kojoj se izvršava glavni program. Kada se program prevede bez grešaka moguće ga je putem USB kabla prebaciti u programsku memoriju mikrokontrolera. Za to je zadužen program koji se već nalazi u delu programske memorije i koji se zove *bootloader*. On zauzima oko 2 kB tako da za skeč aplikaciju preostaje oko 30 kB. Uloga bootladera je da fajl koji je rezultat prevođenja izvornog koda prebaci putem USB porta u programsku memoriju mikrokontrolera nakon čega je moguće izvršiti aplikaciju. Detaljna

objašnjenja, primeri, zadaci i rad u razvojnom okruženju će biti dati na vežbama.



Slika 6.5: Razvojno okruženje Arduino IDE

## 6.2 Arhitektura ATmega328P

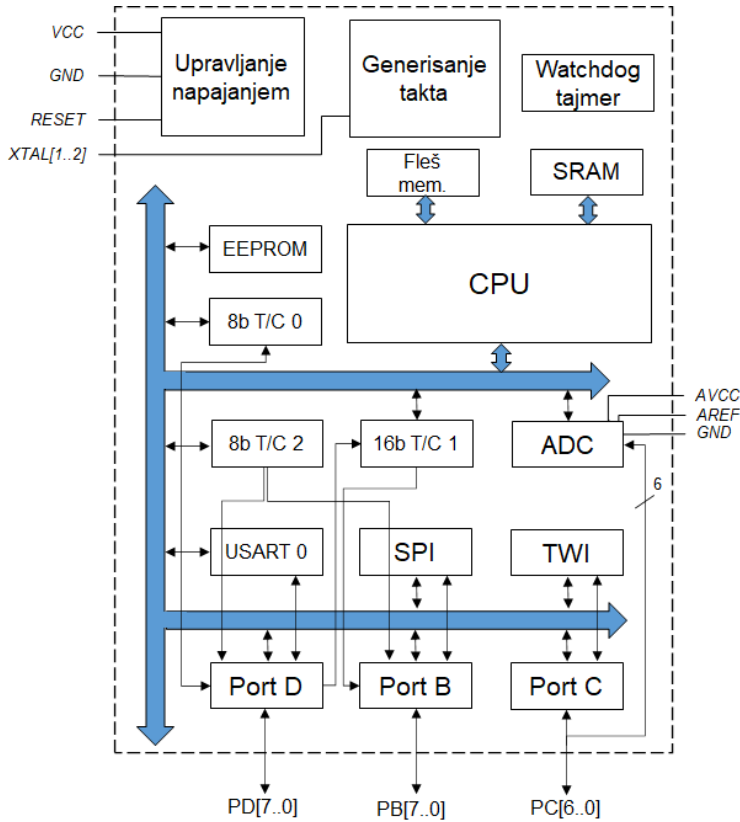
U okviru ove sekcije ćemo se baviti arhitekturom mikrokontrolera ATmega328P. Generalno govoreći, svaki mikrokontroler ima za cilj da radi sa nekim periferijama i to je vrlo često povezano sa radom u realnom vremenu. Na primer, potrebno je uključivati/isključivati neke LED, upravljati brzinom jednosmernog motora, prikazivati nešto na LCD-u i dr. Da bi to mogao, mikrokontroler pored uobičajenih

blokova CPU jezgra, memorija za podatke i program, bloka za generisanje takta i bloka za upravljanje napajanjem, sadrži često dodatne blokove koji olakšavaju rad sa dodatnim periferijama. To su pre svega ulazno/izlazni portovi koji su obično digitalni, ali mogu biti i analogni. Kod ATmega328P postoji tri porta koji su označeni kao Port B, Port C i Port D. Za potrebe komunikacije često postoji bar jedan ili više modula koji implementiraju odgovarajuće serijske komunikacione protokole. Kod ATmega328P imamo tri takva modula. To su USART (eng. *Universal Synchronous/Asynchronous Receiver/Transmitter*), SPI (eng. *Serial Peripheral Interface*) i TWI (eng. *Two Wire Interface*) koji je poznat i pod oznakom I<sup>2</sup>C. Za potrebe merenja vremena, brojanja nekih događaja ili generisanja signala ATmega328P ima tri tajmerska/brojačka modula (8-bitni T/C0, 16-bitni T/C1 i 8-bitni T/C2). T/C je skraćeni zapis engleskih reči *timer/counter*. Za rad sa analognim signalima postoji blok za analogno digitalnu konverziju (ADC). Delimično pojednostavljena arhitektura ATmega328P je prikazana na slici 6.6.

Mikroprocesor unutar mikrokontrolera ima Harvard arhitekturu memorije sa 32 kB programske fleš memorije, 2 kB memorije za podatke (SRAM) i 1 kB EEPROM. Instrukcija se zahvata iz programske memorije u instrukcioni registar (IR) na bazi sadržaja programskog brojača (PC). Upravljački signali se dobijaju na izlazu instrukcionog dekodera (ID). CPU ima registarsko polje od 32x8-bitna registra (GPR). Podržano je direktno i indirektno adresiranje. Arhitektura CPU sa memorijama je prikazana na slici 6.7.

CPU ovog RISC mikrokontrolera ima dvofaznu protočnu obradu. U toku prve faze se vrši zahvat instrukcije (IF), a u drugoj fazi izvršavanje instrukcije (EX). Primer sa četiri instrukcije je prikazan na slici 6.8. Većina aritmetičko-logičkih instrukcija se izvršava u jednom taktu. Na slici 6.9 je prikazan redosled operacija kod izvršenja neke aritmetičko-logičke instrukcije. Pored ukupnog vremena potrebnog za izvršenje instrukcije u trajanju jedne periode takta, prikazane su i podfaze izvršenja instrukcije. Reč je o zahvatu operanda, izvršenju operacije i povratnom upisu.

Registarsko polje sadrži tridest i dva 8-bitna registra opšte namene (R0-R31). Kao što je uobičajeno oni su realizovani kao deo SRAM. Nalaze se na adresama od 0x00-0x1F što je prikazano na slici 6.10. Poslednjih šest adresa (opseg 0x1A-0x1F) koriste se i kao tri specijalna registra u oznaci X, Y, i Z koji su 16-bitni i koji se mogu koristiti za indirektno adresiranje.

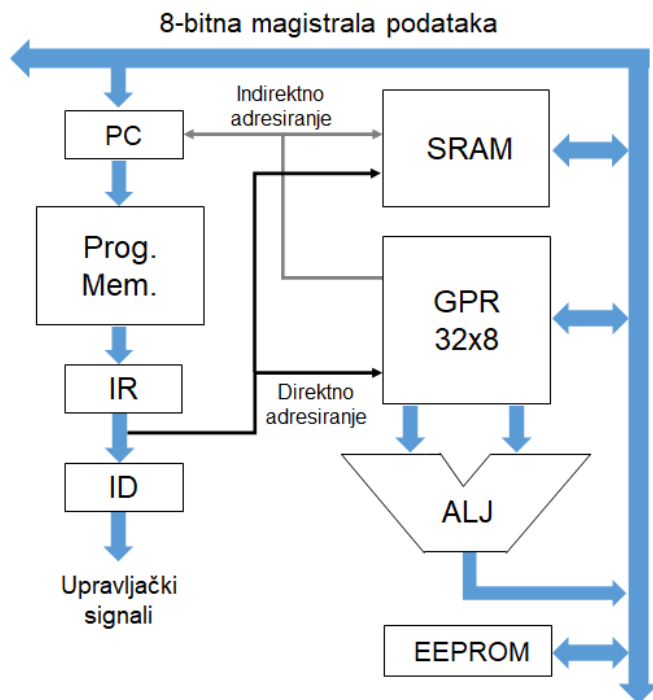


Slika 6.6: Arhitektura mikrokontrolera ATmega328P

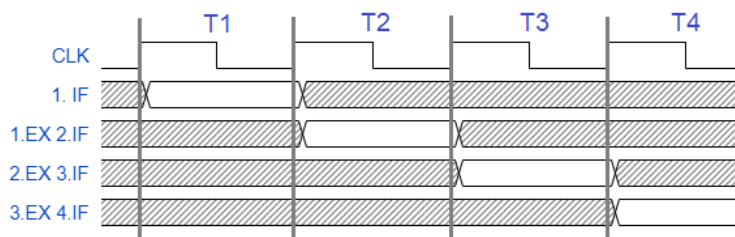
## 6.3 Portovi

ATmega328P ima 3 ulazno-izlazna porta koji su označeni kao Port B, Port C i Port D. Portovi B i D su 8-bitni, a port C je 7-bitni (vidi sliku 6.6). Portovi zauzimaju većinu pinova mikrokontrolera (vidi sliku 6.3). Da bi se obezbedile raznovrsne mogućnosti povezivanja sa periferijama proizvođač je predvideo višestruke alternativne funkcije na skoro svim pinovima. Za kontrolu osnovne funkcionalnosti portova koriste se tri registra. To su  $DDRx$ ,  $PINx$  i  $PORTx$ , pri čemu je  $x \in \{B, C, D\}$  zavisno od konkretnog porta (npr. za Port B se koriste registri  $DDRB$ ,  $PINB$  i



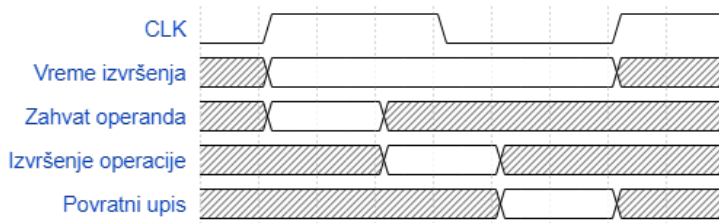


Slika 6.7: Arhitektura CPU ATmega328P sa memorijama



Slika 6.8: AVR dvofazna protočna obrada

PORTB). Registar DDR određuje smer podataka tj. određuje da li je pin porta ulazni ili izlazni. Čitanjem registra PIN se očitava ulazna vrednost sa nekog pina



Slika 6.9: Faze izvršenja aritmetičko-logičkih instrukcija

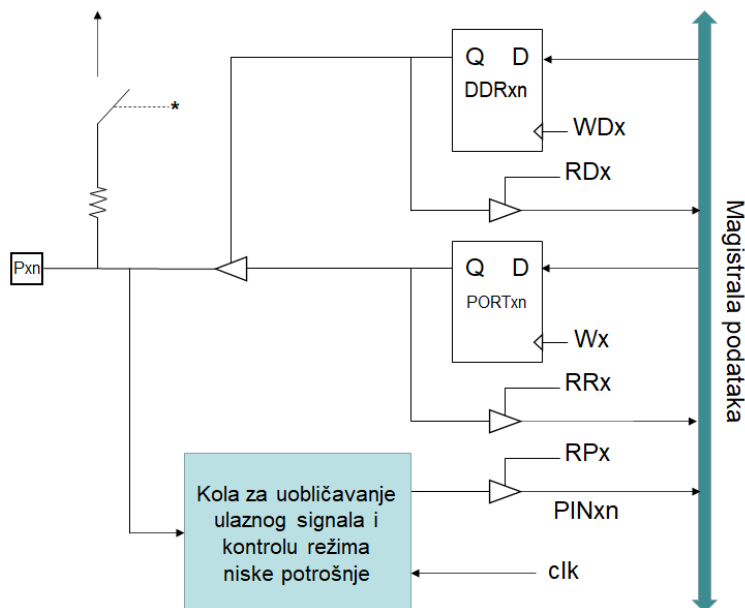
7	0	Adresa	
R0	0x00		
R1	0x01		
R2	0x02		
...			
R13	0x0D		
R14	0x0E		
R15	0x0F		
R16	0x10		
R17	0x11		
...			
R26	0x1A	LSB X registra	
R27	0x1B	MSB X registra	
R28	0x1C	LSB Y registra	
R29	0x1D	MSB Y registra	
R30	0x1E	LSB Z registra	
R31	0x1F	MSB Z registra	

Slika 6.10: Registarsko polje

porta. Na PORT registar se šalje izlazna vrednost za neki pin porta.

Pojednostavljena struktura jednog pina ( $n$ ) porta ( $X$ ) je data na slici 6.11. Na slici važi za svako  $x \in \{B, C, D\}$ , a  $n \in [0 - 7]$ . Svaki pin  $n$  porta  $x$  se može konfigurirati da bude izlazni, ulazni ili ulazni sa *pull-up* otpornikom. Da bi pin  $n$  porta  $x$  bio ulazni potrebno je da se u registar  $DDRx$  na poziciju za pin  $n$  upiše 0 (uz aktivaciju  $WDx$  signala - eng. *write  $DDRx$* ). Tada je isključen bafer na izlazu  $POR-Txn$  flip-flopa i signal sa ulaza pina  $Pxn$  se može čitati preko kola za uobličavanje ulaznog signala i kontrolu režima niske potrošnje ukoliko je aktivan signal  $RPx$  (eng. *read pin  $x$* ). Ukoliko želimo da pročitamo stanje flip-flopa  $PORTxn$  tada je

potrebno da je aktivan signal  $RRx$  (eng. *read register x*). Sa \* je označeno mesto i uslov za uključenje ulaznog *pull-up* otpornika ukoliko je on potreban. Da bi on bio uključen potrebno je da budu ispunjeni sledeći uslovi:  $DDRx=0$  i  $PORTx=1$  (ukoliko je 0, prekidač je isključen). Ukoliko želimo da pin  $n$  porta  $x$  bude izlazni dovoljno je da u  $DDRx$  registar upišemo 1 na poziciju za pin  $n$  (čime uključujemo bafer na izlazu  $PORTxn$ ). Nakon toga upis vrednosti u  $PORTxn$ , uz aktivaciju signala  $W$  (eng. *write*), upisuje željenu vrednost u  $PORTxn$ .

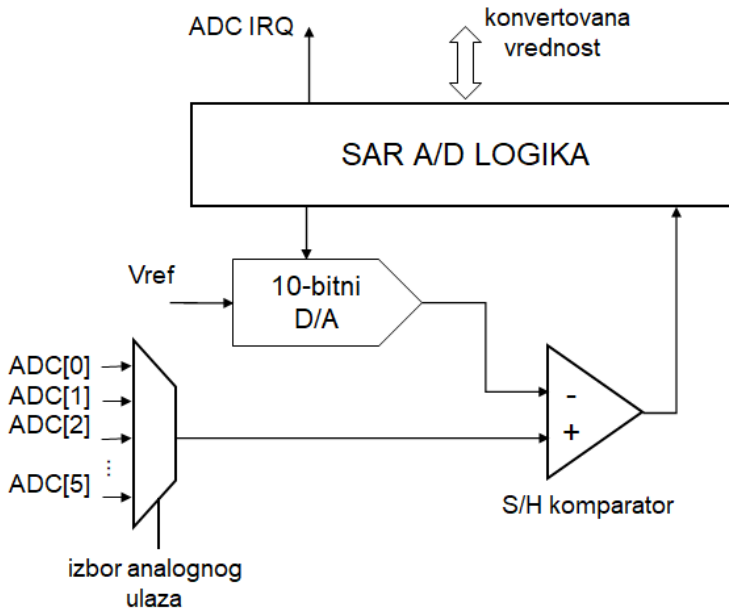


Slika 6.11: Struktura jednog pina porta

## 6.4 Analogni deo

ATmega328P odlikuje mogućnost rada sa analognim signalima. Za tu namenu je opredeljeno šest analognih ulaza u oznaci A0-A5 (vidi sliku 6.3) koji se nalaze na portu C. Sa porta se ovi ulazi vode dalje na analognodigitalni konvertor koji ima zadatak da analogne vrednosti pretvori u digitalne. Koristi se 10-bitni analognodigitalni konvertor koji koristi metodu sukcesivne aproksimacije (SAR

A/D konvertor). Pojednostavljena blok šema ovog A/D konvertora prikazana je na slici 6.12. Vreme konverzije je u opsegu  $65 - 260 \mu s$ . Suština ove metode je da se analogna vrednost na ulazu periodično odabira, unutar konvertora se prethodno sračunata digitalna vrednost pretvara u analognu pomoću D/A konvertora i vodi se na S/H (eng. *sample and hold* komparator gde se porede te dve vrednosti i nastoji se da razlika bude što manja. Smanjivanje razlike se realizuje metodom sukcesivne aproksimacije (polovljenjem). Kada je konverzija završena aktivira se signal zahteva za prekidom (ADC IRQ).



Slika 6.12: Blok šema SAR A/D konvertora

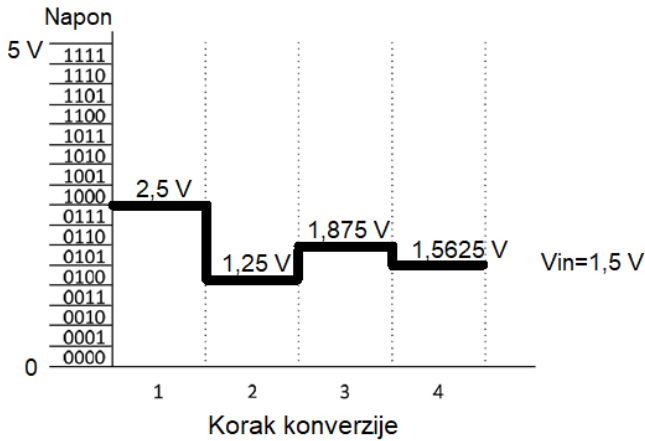
Konvertovana vrednost se nalazi u registarskom paru ADCH:ADCL. Ova vrednost je unutar opsega  $0x000:0x3FF$  (10-bitni opseg) i u idealnom slučaju je linearno zavisna od analognog ulaznog napona  $V_{in}$  prema formuli:

$$ADC = \frac{V_{in} * 1024}{V_{ref}}$$

gde je  $V_{ref}$  referentni napon A/D konvertora. U realnosti se dešavaju greške usled

nelinearnosti, greške kvantizacije i dr.

Metoda sukcesivne aproksimacije ili stalnog približavanja tačnoj vrednosti je ilustrovana na slici 6.13. Ovde je prikazan slučaj 4-bitnog SAR A/D konvertora za ulazni napon  $V_{in}=1,5$  V. Vidi se da je konvertovani napon u svakom narednom koraku konverzije sve bliži ulaznom. Naravno konvertovana vrednost (0x0101 što je ekvivalent 1,5625 V) nije jednaka ulaznoj (1,5 V), ali ovde je reč o ilustraciji sa vrlo niskom rezolucijom od svega 4 bita kod koje je korak kvantizacije (promena na nivou jednog bita) jednak  $5\text{ V}/16=0,3125$ .



Slika 6.13: Ilustracija principa SAR A/D konverzije

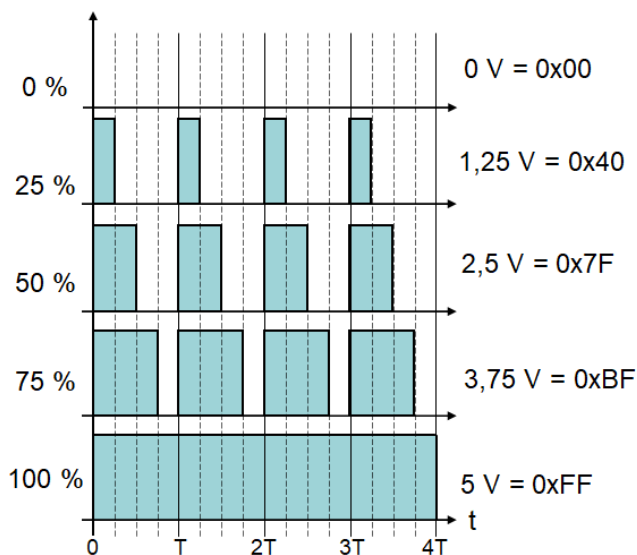
U okviru analognog dela je potrebno spomenuti još i mogućnost realizacije analognih izlaza čija se srednja vrednost može menjati upotrebom PWM-a. Metoda impulsno širinske modulacije se koristi radi modulisanja signala putem faktora ispune (D) koji predstavlja odnos impulsa i pauze unutar periode. Na taj način srednja vrednost napona  $V_{sr}$  je:

$$V_{sr} = D * V_H, D = \frac{T_H}{T} * 100$$

gde je  $T_H$  vreme kada je signal visok unutar periode, a  $T$  je perioda signala.

Modulacijom faktora ispune  $D$  se menja srednja vrednost napona, a time se može menjati srednja vrednost snage izlaznog signala i upravljati određenim uređajima. Na slici 6.14 možemo videti 5 različitih slučajeva modulisanog PWM signala gde je faktor ispune iskazan u procentima 0, 25, 50, 75 i 100 % (na graficima

je obojen sadržaj gde je signal jednak visokom naponskom nivou). Sa desne strane grafika je prikazana i srednja vrednost napona za date slučajeve kao i ekvivalentna 8-bitna vrednost u heksadecimalnoj notaciji. Više reči o radu sa PWM izlazima će biti u poglavlju ?? o tajmerima.



Slika 6.14: Ilustracija impulsno-širinske modulacije

## 6.5 Pitanja

1. Šta je Arduino i zašto se za Arduino ekosistem može reći da je uspešna priča?
2. Navesti osnovne karakteristike Arduinoa.
3. Šta je to *sketch*, koje delove mora da sadrži i čemu oni služe?
4. Data je delimična blok šema mikrokontrolera ATmega328P. Doctati nedostajuće blokove.
5. Data je blok šema mikrokontrolera ATmega328P. Objasniti je.

6. ATmega328P ima ---- flash-a, ---- SRAM-a i ---- EEPROM-a.
7. ATmega328P ima portove: (upisati njihova imena i veličinu) koji su \_\_\_ bitni.
8. ATmega328P ima ---- registara opšte namene koji su \_\_\_ bitni.
9. Kako su realizovani registri opšte namene kod ATmega328P? Po čemu se razlikuju i čemu služe X, Y i Z registri?
10. Objasniti protočnu obradu kod AVR mikrokontrolera.
11. Objasniti načine i faze izvršenja AL instrukcija.
12. Kod ATmega328P, pomoću koja tri registra se upravlja portovima i koji čemu služi?
13. Nacrtati generalnu strukturu portova i objasniti kako se konfiguriše pin porta da bude ulazni/izlazni/sa pull-up otpornikom.
14. Kod ATmega328P, Na koji način se radi sa analognim ulazima, a kako izlazima?
15. Nacrtati blok šemu SAR A/D konvertora i objasniti kako funkcioniše.
16. Objasniti metod sukcesivne aproksimacije na 4-bitnom primeru.
17. Nacrtati i objasniti impulsno-širinsku modulaciju i čemu ona može da služi.

# Bibliografija

- [1] Arthur W. Burks, Herman H. Goldstine, and John von Neumann, "Preliminary discussion of the logical design of an electronic computing instrument", The Institute of Advanced Study, Princeton, USA, 1946-47.
- [2] J. Biggs, J. Myers, J. Kufel et al. A natively flexible 32-bit Arm microprocessor, *Nature* 595, pp. 532–536, 2021.
- [3] F. Arute, K. Arya, R. Babbush et al. Quantum supremacy using a programmable superconducting processor. *Nature* 574, pp. 505–510, 2019.
- [4] J. Hochstetter, R. Zhu, A. Loeffler et al. Avalanches and edge-of-chaos learning in neuromorphic nanowire networks. *Nat Commun* 12, 4008, 2021.
- [5] B. B. Brey, *Intel Microprocessors*, 8th Edition, Pearson, 2009.
- [6] S. Harris, D. Hariss, *Digital Design and Computer Architecture*, Arm edition, Morgan Kaufmann, 2015.
- [7] D. Patterson, J.L. Hennessy, *Computer Organization and Design RISC-V Edition: The Hardware Software Interface*, The Morgan Kaufmann Series in Computer Architecture and Design, 2nd Edition, 2021.
- [8] M. M. Mano, C. R. Kime, T. Martin, *Logic and Computer Design Fundamentals*, Fifth edition, Pearson, 2015.
- [9] R. S. Sandige, M. L. Sandige, *Fundamentals of Digital and Computer Design with VHDL*, McGraw Hill, 2012.
- [10] L. Null, J. Lobur *Essentials of Computer Organization and Architecture*, 5th Edition, Jones and Barlett Publishers, 2018.



- [11] E.O. Hwang, Digital Logic and Microprocessor Design With VHDL with Interfacing, Cengage Learning, Second Edition, 2017.
- [12] I. Mezei "Evolution of an educational microprocessor", Computer Applications in Engineering Education, 28(5), Wiley, pp. 1265-1277, 2020.
- [13] W. Stallings, Computer organization and architecture, 9th Edition, Pearson, 2013.
- [14] M. Abd-El-Barr, H. El-Rewini, Fundamentals Of Computer Organization And Architecture, John Wiley and Sons, Inc., 2005.
- [15] C. Hamacher, Z. Vranesic, S. Zaky, N. Manjikian, Computer Organization and Embedded Systems, 6th Edition, McGraw-Hill, 2012.
- [16] Manuel Jiménez, Rogelio Palomera, Isidoro Couvertier, Introduction to Embedded Systems, Springer, 2014.
- [17] D. A. Patterson, J. L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Fourth Edition, Morgan Kaufmann, 2011.
- [18] J.L. Gustafson, Moore's Law. In: D. Padua D. (eds) Encyclopedia of Parallel Computing. Springer, 2011.
- [19] S.A. McKee, R.W. Wisniewski, Memory Wall. In: D. Padua (eds) Encyclopedia of Parallel Computing. Springer, 2011.
- [20] P. Bose, Power Wall. In: D. Padua (eds) Encyclopedia of Parallel Computing. Springer, 2011.
- [21] A. Tsakyridis, T. Alexoudi, A. Miliou, N. Pleros, and C. Vagionas, "10 Gb/s optical random access memory (RAM) cell," Opt. Lett. 44, pp. 1821-1824, 2019.
- [22] W.W. Royce, "Managing the Development of Large Software Systems: Concepts and Techniques", IEEE Wescon Proc., Aug. 1970.
- [23] C.G. Bell, A. Newell, Computer structures: Readings and Examples, McGraw-Hill, 1971.
- [24] S. Brown, Z. Vranesic, Fundamentals of Digital Logic with VHDL Design, McGraw Hill, 2009.
- [25] D. L. Perry, VHDL: Programming by Example, Fourth edition, McGraw Hill, 2002.

- [26] L. Null, J. Lobur, Essentials of Computer Organization and Architecture, Jones and Bartlet Learning, 3rd Ed., 2010.
- [27] D. Patterson, J.L. Hennessy, Computer Organization and Design RISC-V Edition: The Hardware Software Interface, The Morgan Kaufmann Series in Computer Architecture and Design, 1st Edition, 2017.
- [28] S. L. Harris, D. Harris, Digital Design and Computer Architecture, RISC-V Edition, Morgan Kaufman, 1st edition, 2021.
- [29] D. Patterson, "Reduced Instruction Set Computers Then and Now" in Computer, vol. 50, no. 12, pp. 10-12, 2017. <https://doi.ieeecomputersociety.org/10.1109/MC.2017.4451206>
- [30] D. A. Patterson, J. L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, ARM edition, The Morgan Kaufmann Series in Computer Architecture and Design, 2016.
- [31] D. Kushner, "The making of arduino", IEEE Spectrum 26, 2011.
- [32] M. Banzi, "How Arduino is open-sourcing imagination", TEDtalk, Scotland, 2012. <https://www.youtube.com/watch?v=UoBUXOOdLXY> (datum pristupa: 31.01.2022.)
- [33] Sparkfun, "Arduino shields v2", <https://learn.sparkfun.com/tutorials/arduino-shields-v2>, (datum pristupa: 01.02.2022.)
- [34] Arduino Uno Rev3 schematic, [https://www.arduino.cc/en/uploads/Main/Arduino\\_Uno\\_Rev3-schematic.pdf](https://www.arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf) (datum pristupa: 02.02.2022.)
- [35] Arduino software, <https://www.arduino.cc/en/Main/Software> (datum pristupa: 02.02.2022.)
- [36] 74HC595 datasheet, Philips, 1998, <https://www.arduino.cc/en/uploads/Tutorial/595datasheet.pdf> (datum pristupa: 03.03.2022.)
- [37] <https://www.ti.com/lit/ds/symlink/sn74hc165.pdf> (datum pristupa: 03.03.2022.)
- [38] ATmega328 datasheet, Atmel, 2015, [https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf) (datum pristupa: 15.03.2022.)

- [39] Tutorials 83, Fundamentals of RS-232 serial communications, Analog Devices, 2001. <https://www.maximintegrated.com/en/design/technical-documents/tutorials/8/83.html> (datum pristupa: 31.03.2022.)