

Akvizicija i vizuelizacija naponskih signala pomoću Arduino mikrokontrolera i PC računara zasnovana na primjeni Python biblioteka

Studentski rad

Dorđe Kovačević
 Student prvog ciklusa studija
 Univerzitet u Istočnom Sarajevu, Elektrotehnički
 fakultet
 Istočno Sarajevo, Bosna i Hercegovina
 djordje.kovacevic.2245@student.etf.ues.rs.ba

Sažetak — U ovom radu razmatra se primjena Python biblioteka u sistemima za akviziciju i vizuelizaciju naponskih signala zasnovanim na Arduino mikrokontrolerima. Predstavljen je jednostavan mjerni sistem u kojem se analogni naponski signali prenose sa Arduino Mega 2560 razvojne ploče do PC računara putem USB serijske komunikacije i dalje obrađuju korišćenjem Python programskog jezika. Analizirane su mogućnosti i kompatibilnost odabranih Python biblioteka za serijsku komunikaciju, obradu podataka i grafički prikaz mjernih signala. U eksperimentalnom dijelu rada, prikazana je primjena Arduino Mega 2560 mikrokontrolera za mjerenje napona trofaznog naponskog sistema, uz vizuelizaciju naponskih talasnih oblika u Python okruženju. Rezultati pokazuju da ovakav pristup predstavlja jednostavno i ekonomično rješenje pogodno za edukaciju i laboratorijske primjene.

Cljučne riječi: akvizicija podataka; Arduino Mega 2560; Python biblioteka; serijska komunikacija; vizuelizacija signala

I. UVOD

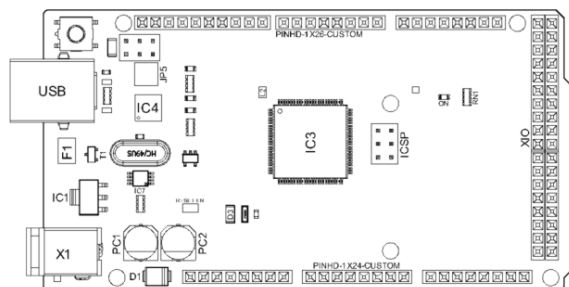
Sa napretkom i ekspanzijom računarske tehnike, personalni računari (PC, Personal Computer) u savremenim naučno-istraživačkim tokovima integrisani su u okvire širokog spektra novih primjena. Jedna od najznačajnijih primjena računara počiva u sistemima specijalizovanim za akviziciju i kontrolu podataka (Data Acquisition and Control Systems), koji služe za rješavanje izuzetno složenih problema u kontekstu savremenih naučnih istraživanja [1]. Zamjenom starih, namjenskih uređaja, koji su se primjenjivali u laboratoriji i procesnoj industriji, personalnim računarima, usavršeni su visokointeligentni sistemi sa značajno poboljšanim upravljačkim karakteristikama, što je za direktnu posljedicu imalo dobijanje relevantnijih rezultata, kvalitetnijih zaključaka, kao i nižu cijenu izrade i održavanja ovakvih sistema. Pored navedenog, važno je istaći i ulogu mikrokontrolera u sistemima za akviziciju i obradu podataka, gdje se gotovo uvijek mogu sresti zbog svoje niske cijene i multifunkcionalnosti (omogućavaju precizno određivanje vremenskih intervala, mjerenje trajanja vanjskih događaja, kao i njihovo brojanje) [2] [3] [4]. U okviru ovog rada, za akviziciju i vizuelizaciju podataka korišten je programski jezik Python, zbog sveobuhvatnih funkcionalnosti i intuitivne sintakse. Iako njegova upotreba iziskuje dodatne napore

usmjerene ka algoritamskoj optimizaciji i očuvanju performansi, Python nudi efikasne i robusne biblioteke koje značajno olakšavaju proces rada [5]. U okviru eksperimentalnog dijela razvijen je sistem koji uspješno prikuplja, obrađuje i grafički predstavlja signale sa trofaznog naponskog izvora.

II. ARDUINO MIKROKONTROLERI U SISTEMIMA ZA AKVIZICIJU NAPONSKIH SIGNALA

Arduino ploče koriste različite vrste mikrokontrolera i opremljene su setovima analognih i digitalnih ulazno-izlaznih pinova, a, takođe, mogu se povezivati sa različitim elektronskim kolima i sklopovima, poput ploča za izradu prototipova ili ploča za proširivanje (shield). Ploče imaju serijske komunikacione interfejse, uključujući i zaseban USB modul koji ujedno služi i za učitavanje programa.

Sa softverskog aspekta, Arduino mikrokontroleri predstavljaju vrlo često i pouzdano rješenje u različitim sistemima za akviziciju podataka i u svrhe izrade i učitavanja koda realizovano je posebno okruženje – Arduino IDE (Integrated Development Environment). Arduino IDE je multiplatformska aplikacija zasnovana na Processing IDE, koji je napisan u programskom jeziku Java. Ovo okruženje podržava rad sa programskim jezicima C i C++, ali važno je napomenuti da sam program za Arduino hardver može biti napisan u bilo kom programskom jeziku sa kompajlerom koji generiše binarni mašinski kod za unaprijed određeni tip mikrokontrolera.



Slika 1. Unutrašnja struktura mikrokontrolera Arduino Mega 2560

Predstavljeni mikrokontroler posjeduje 4 interfejsa za serijsku komunikaciju (UART), USB konektor, 16 MHz

kristalni oscilator i dugme za resetovanje, što dati mikrokontroler čini pogodnom solucijom za implementaciju u okviru različitih akvizicionih sistema. Posebno je interesantna upotreba Arduino mikrokontrolera u sistemima za akviziciju i skladištenje naponskih signala, budući da je jedna od njihovih gradivnih i funkcionalnih komponenata AD konvertor (Analog-to-digital converter).

AD konvertor je sistemska komponenta koja pretvara kontinualne, analogne signale u diskretne, digitalne signale, a cjelokupna funkcionalnost ovog modula ostvarena je preko ADC kanala (A0-A15), koji vrše konverziju u desetobitnoj rezoluciji (tj. u opsegu vrijednosti od 0 do 1023). Pomoću opisanog konvertora moguće je mjeriti jednosmjerne napone čija amplituda varira u opsegu od 0 do 5 V.

Proces konverzije traje u rasponu od 100 do 200 μ s, a podrazumijevana referentna vrijednost je 5 V, mada postoji mogućnost njenog redefinisavanja, koje se ostvaruje tako što se novi referentni napon dovede na pin „AREF”, a u kodu se pozove funkcija *analogReference()* kojoj će se proslijediti parametar „EXTERNAL” ili se, alternativno, odabere unutrašnja referenca, kada se pomenutoj funkciji prosljeđuje parametar „INTERNAL”.

ADC pinovima se softverski pristupa pomoću funkcije *analogRead()* koja mapira ulazne naponske signale linearno u cjelobrojne vrijednosti (tip podatka: integer). Minimalna naponska varijacija koju konvertor može detektovati izražava se kao količnik maksimalne vrijednosti napona koji je moguće registrovati na ulaznim pinovima i broja diskretnih naponskih nivoa:

$$\frac{5 \text{ V}}{1024} \approx 4.88 \text{ mV.} \quad (1)$$

Ulazni naponi koji premašuju vrijednost od 5 V potencijalno bi mogli da izazovu oštećenja, pa se, shodno tome, ukoliko postoji potreba za mjerenjem naponskih signala veće vrijednosti, vrši translacija ulaznog napona na navedeni opseg od 0 do 5 V.

Na mikrokontrolerskoj pločici nalazi se USB interfejs tipa B preko kog je ostvarena direktna komunikacija između mikrokontrolera koji vrši konverziju i mjerenje naponskih signala i računara koji prikuplja digitalizovane podatke i vizualizuje ih. Za serijsku komunikaciju koristi se USB serijski procesor ATmega16U2 koji radi na frekvenciji od 16 MHz.

III. PYTHON BIBLIOTEKE ZA AKVIZICIJU I OBRADU PODATAKA

Python je programski jezik visokog nivoa opšte namjene koji je stekao veliku popularnost i primjenu u savremenim softverskim rješenjima zbog svoje jednostavne, intuitivne sintakse, kao i zbog toga što podržava različite paradigme programiranja, što pruža veliku fleksibilnost u radu. Ovaj programski jezik koristi se za izradu različitih aplikacija u različitim granama industrije – od razvoja video-igara do poslovnih veb-aplikacija [6].

Posmatrano iz ugla primjene u svrhu naučno-tehnoloških istraživanja, Python se pokazao kao jedno od najoptimalnijih rješenja za primjenu u sistemima za akviziciju i obradu podataka. Postoji nekoliko razloga koji opravdavaju njegovu visoku prisutnost u sistemima tog tipa: veliki broj biblioteka koje olakšavaju rad sa različitim tipovima podataka, velika zajednica aktivnih korisnika, kvalitetna korisnička

dokumentacija, prenosivost koda. Brojni gotovi moduli omogućavaju rješavanje problema bez pisanja koda „od nule”.

U okviru ovog rada primarni fokus će biti stavljen na nekoliko biblioteka čije funkcije su u najvećoj mjeri korištene u realizaciji eksperimentalnog dijela:

A. PySerial

Biblioteka PySerial omogućava pristup serijskom portu i na osnovu funkcionalnosti koje enkapsulira ostvarena je komunikacija između Arduino mikrokontrolera i Python skriptata. PySerial posjeduje nekoliko značajnih karakteristika koje značajno olakšavaju rad:

- Isti interfejs na svim podržanim platformama.
- Pristup podešavanjima porta posredstvom Python svojstava (properties).
- Podrška za različite veličine paketa, stop-bitove, bitove pariteta i kontrolu protoka.
- Rad sa vremenskim ograničenjem prijema ili bez njega.
- Port je podešen za binarni prenos [7].

B. Matplotlib

Python biblioteka Matplotlib pruža korisniku mogućnost kreiranja statičkih, animiranih i interaktivnih vizuelizacija. Set njenih funkcionalnosti je sveobuhvatan, što podrazumijeva da je moguće mijenjati sve aspekte vizuelizacije (izgled grafičke reprezentacije prikupljenih podataka, oznake osa, hromatski spektar, skaliranje, raspored figura) [8].

C. NumPy

NumPy je Python biblioteka koja posjeduje šarenolik skup matematičkih funkcija koje su naročito pogodne za obradu prikupljenih podataka. U okviru ove biblioteke implementirani su generatori nasumičnih brojeva, rutine linearne algebre i Furijeove transformacije. Ova biblioteka obiluje brzim i efikasnim funkcionalnostima vektorizacije, indeksiranja i emitovanja, što predstavlja jedan od osnovnih standarda savremenog računarstva [9].

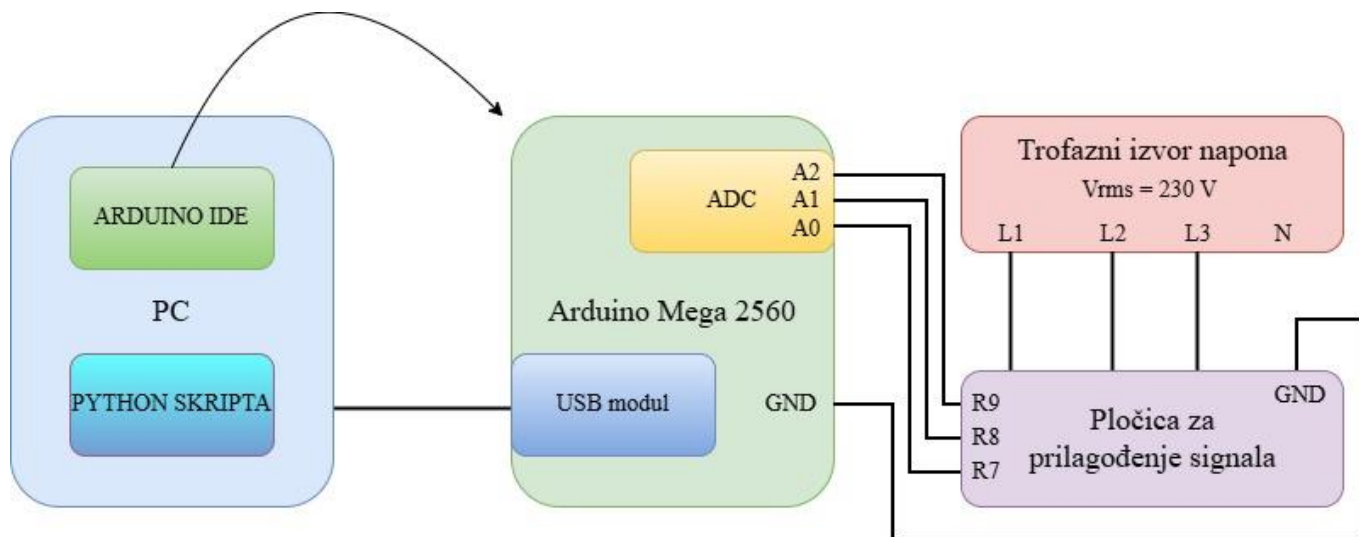
D. Time

Ovaj modul pruža različite funkcije koje su u vezi sa realnim vremenom. Iako je sama biblioteka uvijek dostupna, njene funkcionalnosti zavise od platforme na kojoj se program izvršava. Većina njenih funkcija poziva funkcije C biblioteke sa istim imenom, koja se nalazi na istoj platformi. Upravo ova C biblioteka sadrži tabelu u kojoj su pohranjeni podaci o lokalnim pravilima (računanje vremena, vremenska zona), što pruža fleksibilnost i pouzdanost u radu sistema [10].

E. Csv

Python Csv biblioteka implementira funkcije koje omogućavaju čitanje i upis tabelarnih podataka u csv formatu. Takođe, data biblioteka pruža korisniku mogućnost definisanja sopstvenog csv formata koji će se koristiti za razmjenu informacija sa drugim aplikacijama.

Takozvani csv (Comma Separated Values) format je najčešći format za izvoz i uvoz podataka u tabele i baze podataka, što programskom jeziku Python daje veliku prednost u primjeni u okviru akvizicionih sistema, jer olakšava rad na pohranjivanju i manipulaciji podacima [11].



Slika 2. Konceptualni dijagram realizovanog sistema za akviziciju i vizuelizaciju naponskih signala

IV. ARHITEKTURA SISTEMA ZA AKVIZICIJU I VIZUELIZACIJU NAPONSKIH SIGNALA

Na Sl. 2 predstavljen je konceptualni dijagram sistema za akviziciju i vizuelizaciju naponskih signala realizovanog u okviru ovog rada. Okosnicu cjelokupnog sistema čine 4 fundamentalna segmenta: Arduino mikrokontroler, USB komunikacija, personalni računar i Python program.

A. ARDUINO IDE okruženje

Program koji je učitana na mikrokontroler i koji će se konstantno izvršavati u toku procesa mjerenja, akvizicije i vizuelizacije podataka napisan je u okruženju Arduino IDE, koje je specijalizovano za kreiranje programa namijenjenih za istoimene mikrokontrolere. Sam program strukturisan je tako da posjeduje dvije osnovne funkcije: *setup()* i *loop()*.

U funkciji *setup()* vrši se inicijalizacija i određivanje vrijednosti parametara koji su neophodni za ispravno funkcionisanje programa, ali i cjelokupnog sistema i vrlo je važno istaći da će se sve naredbe napisane u okviru ove funkcije izvršiti samo jedanput i to neposredno prije ulaska u glavnu petlju. U okviru predstavljene funkcije prvo je definisano da će se komunikacija između mikrokontrolera i personalnog računara odvijati pri brzini od 500 000 baud/s, pomoću funkcije *Serial.begin()*. Takođe, definiše se određeno vrijeme čekanja pomoću funkcije *delay()*. Svrha ovog čekanja leži u činjenici da je cjelokupnom sistemu potrebno vrlo kratko vrijeme kako bi se stabilizovao i pripremio za prenos podataka.

Čitanje podataka sa AD konvertora realizovano je preko korisnički kreirane funkcije *read-adc-stable()*. Predstavljena funkcija u sebi sadrži poznatu funkciju *analogRead()* koja predstavlja jezgro njene funkcionalnosti, ali data funkcija se u njoj javlja dvaput. Ovaj koncept je poznat kao tzv. „dummy read“, a zasniva se na jednom čitanju čija vrijednost neće biti zabilježena, kako bi se omogućila stabilizacija vrijednosti na pinu i dobijanje pouzdanih očitavanja.

Nakon prikupljanja vrijednosti sa odgovarajućih pinova, podatke je neophodno pomoću serijske komunikacije prosljediti personalnom računaru, tj. Python programu pokrenutom na njemu, što se postiže funkcijom *Serial.print()*. Takođe, u okviru programa koji se učitava na mikrokontroler

definisana je i frekvencija uzorkovanja koja, u konkretnom slučaju, iznosi 1 kHz, što omogućava veoma pouzdanu rekonstrukciju originalnog mjenog signala, budući da je njegova frekvencija, za posmatrani slučaj, jednaka 50 Hz. Važno je istaći da se podaci ka računaru šalju kao cjelobrojne vrijednosti pristigle sa AD konvertora, a da se njihova konverzija i preračunavanje obavlja u okviru Python skriptata.

B. Python skripta

U okviru programa koji se izvršava na računaru definišu se, analogno prethodnom dijelu, parametri koji određuju način odvijanja serijske komunikacije – brzina prenosa i broj kanala, kao i, u slučaju pohranjivanja podataka, naziv csv fajla.

Prije glavne petlje programa izvršeno je nekoliko podešavanja:

- Alokacija memorijskog prostora za skladištenje podataka: Podaci se skladište u memorijsku strukturu *deque*.
- Podešavanje grafika: Budući da se prikazuju dva grafika – jedan za prikaz napona koji dolaze do mikrokontrolera i drugi za prikaz originalnih, trofaznih napona, neophodno je izvršiti formatiranje odgovarajućih grafika, podešavanje osa, imenovanje vertikalne i horizontalne ose i slično. Ovi zahtjevi postižu se upotrebom korisnički definisanih funkcija: *calculate-adaptive-ylim()* (automatsko računanje granica vertikalne ose), *update-statistics-display()* (ažuriranje grafika novim vrijednostima mjerenja), *calculate-statistics()* (izračunava osnovne statističke karakteristike signala).
- Preračunavanje vrijednosti: Neophodno je rekonstruisati originalne signale (sa izvora i mikrokontrolera) na osnovu vrijednosti dobijenih sa AD konvertora, te je potrebno izvršiti njihovo preračunavanje u odnosu na odgovarajuće amplitude.

U glavnoj petlji programa poziva se funkcija *process-serial-data()* koja omogućava prihvatanje podataka sa serijskog porta.

C. USB komunikacija

Kompletan tok USB komunikacije ostvaren je na osnovu već pomenutih funkcija: `Serial.print()` iz Arduino IDE okruženja i `process-serial-data()` iz Python koda.

Funkcija `Serial.print()` šalje podatke na serijski port u formi ASCII teksta čitljivog ljudima. Brojevi se kodiraju tako što se svaka cifra mijenja odgovarajućim ASCII kodom. Decimalni brojevi se obično zaokružuju na dvije decimale, a bajtovi se šalju kao jedan znak. Specijalni simboli i stringovi se šalju u svojoj originalnoj formi.

Funkcija `process-serial-data()` otvara odgovarajući COM port, podešava brzinu prenosa i čisti ulazne bafer. Takođe, pomenuta funkcija prikuplja podatke pozivom sljedeće metode: `ser.readline()` na koju se nadovezuju dodaci za dekodiranje i uređivanje podataka: `decode()` i `strip()`, sa odgovarajućim argumentima.

V. EKSPERIMENTALNA POSTAVKA

Prilikom izvođenja eksperimentalnog dijela korištena je već pomenuta hardverska oprema, koja je naznačena na Sl. 2: mikrokontroler Arduino Mega 2560, izvor trofaznog naponskog signala, prilagodna ploča, personalni računar i medijumi za komunikaciju i prenos podataka. Rad na računaru se odvijao u okviru okruženja Arduino IDE i okruženja PyCharm u kom je napisan program za vizuelizaciju naponskih signala uz upotrebu programskog jezika Python.

Prvi korak u realizaciji sistema za akviziciju i vizuelizaciju naponskih signala bilo je adekvatno dovođenje trofaznih naponskih signala od izvora do mikrokontrolera. Budući da vrijednosti mjenog napona značajno premašuju opseg ulaznih napona na analognim ulazima (pinovima) mikrokontrolera, bilo je neophodno izvršiti prilagođenje naponskih signala i njihovu translaciju u dati vrijednosni opseg. Dato podešenje ostvareno je posredstvom prilagodne ploče, koja predstavlja elektronski sklop za snižavanje vrijednosti napona, čiji rad se bazira na primjeni transformatora, otporničkog djelatnika napona i jednosmjernog napona vrijednosti 5 V. Svaka od naponskih faza dovedena je na odgovarajući ulaz prilagodne ploče, gdje se, potom, snižava na vrijednost u okviru željenog opsega i sa naznačenih izlaza odvodi ka mikrokontroleru koji može korektno da ih obradi. Oznake ulaza i izlaza prikazane su na Sl. 2.

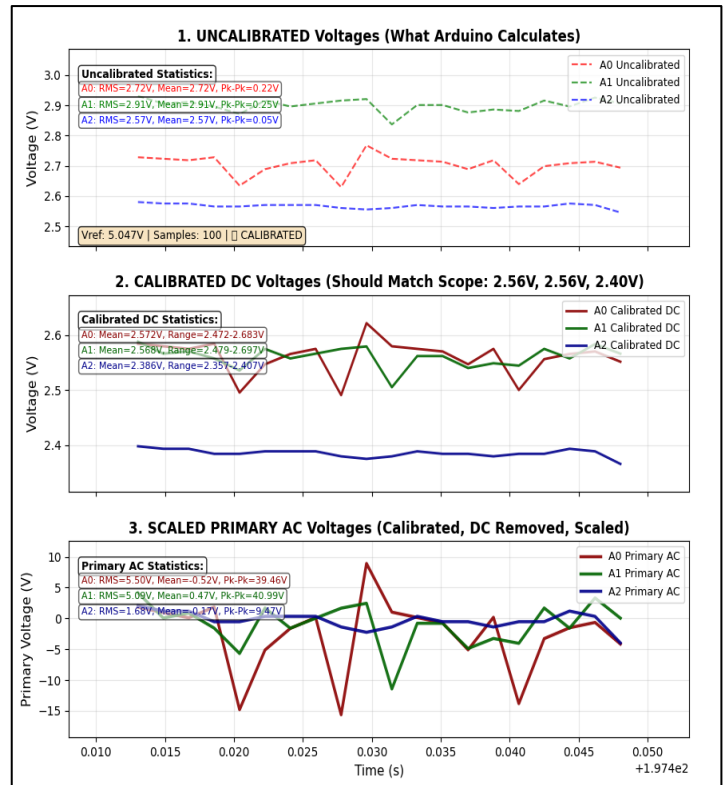
Korektnost i kvalitet rada akvizicionog sistema počiva na frekvenciji uzorkovanja posmatranog trofaznog signala od strane mikrokontrolera, što se definiše softverski, u okviru programa napisanog u okruženju Arduino IDE. Prema teoremi o odabiranju, frekvencija uzorkovanja mora biti minimalno dvostruko veća od frekvencije signala koji se uzorkuje:

$$f_o \geq 2f_s. \quad (2)$$

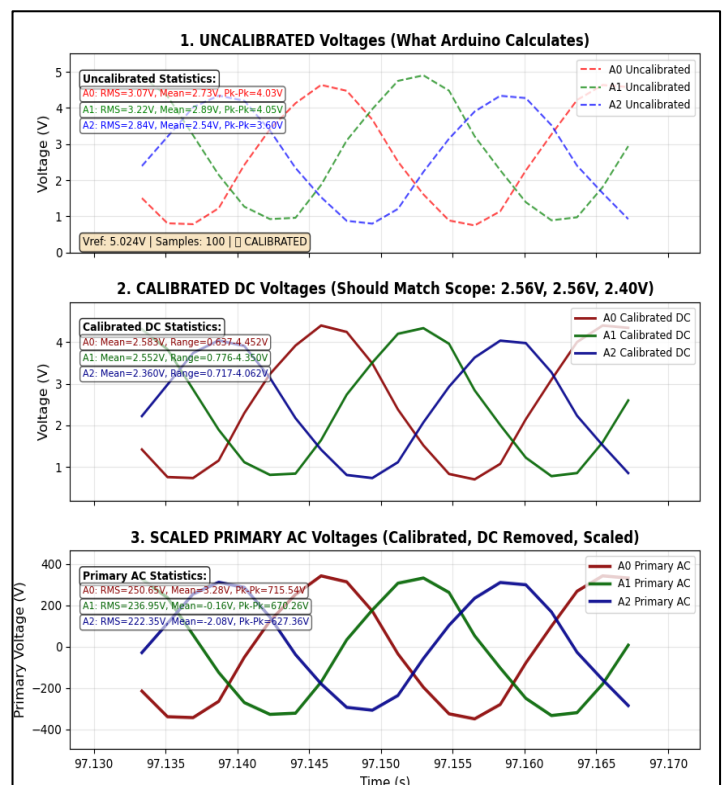
U konkretnom slučaju, frekvencija posmatranog signala iznosi 50 Hz, dok je frekvencija uzorkovanja 1000 Hz, što višestruko prevazilazi minimalne okvire i ima pozitivno dejstvo na funkcionisanje sistema, jer pruža vjerodostojniji i pouzdaniji vizuelni prikaz originalnih signala.

Nadalje, ploča mikrokontrolera je putem USB kabela povezana sa računarnom, čime je osigurana nesmetana komunikacija. Na ovaj način od mikrokontrolera ka računaru, tj. programu za vizuelizaciju prenose se očitavanja sa AD konvertora na osnovu kojih se vrši analiza signala i njegova grafička rekonstrukcija u okviru Python programa.

U okviru predstavljenog eksperimenta dobijeni su sljedeći rezultati prikazani u formi grafika:



Slika 3. Prikaz vrijednosti nekalibrisanog DC, kalibrisanog DC i skaliranog AC napona prije uključjenja trofaznog izvora napona



Slika 4. Prikaz vrijednosti nekalibrisanog DC, kalibrisanog DC i skaliranog AC napona nakon uključjenja trofaznog izvora napona

Takođe, mjere predostrožnosti moraju biti jedan od fundamentalnih elemenata pri realizaciji ovakvih vrsta sistema, budući da se proces mjerenja i obrade vrši nad naponima visoke amplitude koji potencijalno mogu biti

veoma opasni. Vrlo je važno osvrnuti se na aspekt sigurnosti u radu prilikom izvođenja eksperimenta, što podrazumijeva nekoliko ključnih tačaka:

- Galvanska izolacija: Potrebna zaštita ostvarena je u vidu galvanske izolacije na prilagodnoj ploči.
- Referentni napon: Kako bi mjerenja bila pouzdana, potrebno je obezbijediti da mikrokontroler i prilagodna ploča dijele istu tačku uzemljenja kako ne bi došlo do pojave prenapona ili tzv. „plivajuće” mase.
- Strujno ograničenje: Budući da su analogni ulazi namijenjeni za mjerenje napona, poželjno je da struja bude što manja, optimalno ispod 1 mA.

VI. DISKUSIJA REZULTATA

A. Evaluacija rezultata prije uključenja trofaznog naponskog izvora

Na Sl. 3 prikazani su grafici na kojima su predstavljeni talasni oblici naponskih signala, koji dolaze do mikrokontrolera, prije uključenja trofaznog naponskog izvora.

Na prvom grafiku predstavljene su trenutne vrijednosti napona koje mikrokontroler preračunava na osnovu vrijednosti koje dobija od strane AD konvertora, a koje se računaju skalirane u odnosu na nominalni napon od 5 V. Predstavljene signali nisu skalirani, niti kalibrisani što za posljednju ima najveću grešku mjerenja u ovom slučaju.

Drugi grafik oslikava kalibrisane DC napone i vizuelno je vrlo lako uočljivo da su očitane vrijednosti veoma bliske (raspon od 2.4 V do 2.6 V). Signali su kalibrisani pomoću referentnih vrijednosti dobijenih mjerenjem pomoću osciloskopa, čime je softverski umanjena greška mjerenja.

Na trećem grafiku u okviru Sl. 3 mogu se uočiti krive koje predstavljaju rekonstrukciju primarnog, naizmjeničnog napona u slučaju kada trofazni naponski izvor nije uključen. Svi signali su prikazani uz eliminaciju jednosmjernje komponente.

B. Evaluacija rezultata nakon uključenja trofaznog naponskog izvora

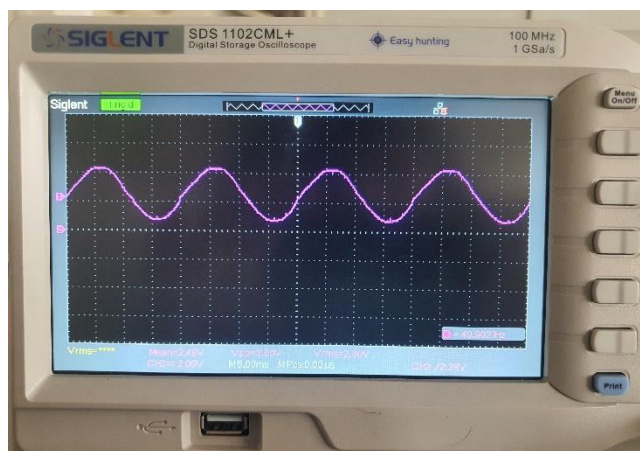
Na Sl. 4, kao i u prethodnom slučaju, predstavljana su 3 grafika, na kojima su prikazani naponski signali translirani u opseg od 0 do 5 V, kao i finalni grafik na kojim je izvršen prikaz originalnih trofaznih naponskih signala.

Već na prvom grafiku je evidentno da je fazni pomak između različitih faza jednak i konzistentan, što pokazuje stabilnost rada izvora, ali i pouzdanost sistema koji dosljedno oslikava obrađene signale. U okviru ovog grafika predstavljeni su talasni oblici napona koje preračunava sam Arduino mikrokontroler na osnovu vrijednosti dobijenih od strane AD konvertora.

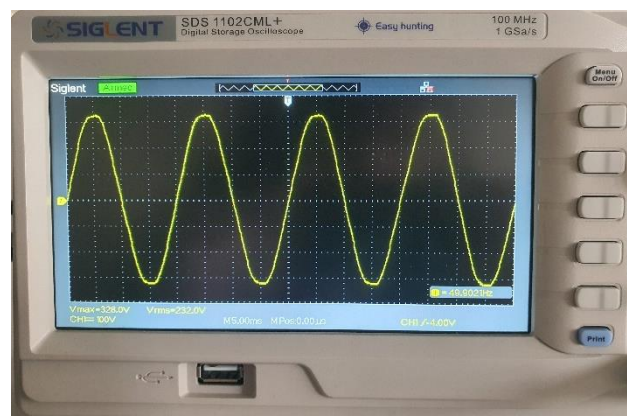
Na drugom grafiku prikazani su kalibrisani naponski signali i važno je zapaziti da su sva tri signala prilično usaglašena sa aspekta statističkih parametara.

Treći grafik na Sl. 4 prikazuje originalne trofazne naponske signale vizualizovane pomoću vrijednosti dobijenih očitavanjem mikrokontrolera i softverskim preračunavanjem u okviru Python programa, što, ujedno, predstavlja i osnovni cilj ovog rada. Nakon kalibrisanja i uklanjanja jednosmjernje komponente, sa grafika se može zaključiti da je očuvana simetrija u odnosu na srednju vrijednost, kao i vrijednosti

amplituda. Ukoliko se uzme u obzir da je riječ o faznom naponu efektivne vrijednosti 230 V, može se zaključiti da grafik za potrebe edukacione analize poprilično vjerno oslikava naponske signale. Korekciju i minimizaciju grešaka je moguće ostvariti dodatnom kalibracijom koja bi mogla biti jedan od potencijalnih ciljeva nastavka rada na sistemu.



Slika 5. Prikaz napona sa jednog od analognih ulaza mikrokontrolera na osciloskopu



Slika 6. Prikaz napona jedne faze trofaznog sistema na osciloskopu

Komparacijom grafika predstavljenih na Sl. 5 i 6 sa odgovarajućim graphicima predstavljenim u okviru Sl. 3 i 4, zaključuje se da postoji međusobna podudarnost što potvrđuje hipotezu o ispravnosti funkcionisanja realizovanog sistema. Budući da je osciloskop uređaj koji ima mogućnost da prikazuje različite vrste signala sa širokim amplitudskim spektrom, ovakva vrsta poređenja korištena je kao validacija kvaliteta sistemskog rješenja i svojevrsan etalon za tumačenje dobijenih grafika.

VII. ZAKLJUČAK

Na osnovu svih predstavljenih činjenica, moguće je izvesti zaključak da predstavljeni sistem za akviziciju i vizuelizaciju naponskih signala trofaznog sistema radi pouzdano, konzistentno i bezbjedno. Provjerom pomoću osciloskopa, utvrđeno je da dobijeni oblik talasnih signala vjerodostojno oslikava trofazne signale sa naponskog izvora, čime je izbor frekvencije uzorkovanja signala potvrđen kao korektan.

Takođe, važno je istaći značaj Python biblioteka koje su se pokazale kao vrlo pouzdan alat u izradi softverskog rješenja za vizuelizaciju signala, jer nude širok spektar funkcija koje značajno olakšavaju rad sa vremenskim i tehničkim aspektima posmatranja. Na predstavljenom primjeru pokazano je da se

upotrebom Python biblioteka može izrisiti relativno pouzdano i efikasno rješenje za vizuelizaciju signala u edukacione svrhe.

Cjelokupan sistem realizovan je u potpunosti pomoću komponenata i uređaja dostupnih u laboratoriji, iz čega se može zaključiti da je ovakvo akviziciono rješenje jeftino za realizaciju i vrlo kvalitetno za primjenu u svrhu edukacije i istraživanja. Ovakav sistem može poslužiti kao prototip za dalju nadogradnju, usavršavanje i razvoj sličnih sistema koji bi mogli da obrađuju naponske signale viših amplituda ili učestanosti. U toku realizacije ovakvog sistema akcenat nije bio na vremenskom aspektu obrade podataka, ali važno je naglasiti da je krajnja akvizicija i vizuelizacija signala poprilično bliska obradi u realnom vremenu. Uzevši u obzir sve navedene parametre, realizovani sistem za akviziciju i realizaciju se može smatrati kvalitetnim i efikasnim.

Sa druge strane, uočeno je i nekoliko segmenata koji mogu poslužiti kao smjernica za poboljšanje tehničkih performansi sistema u nastavku razvoja:

- Algoritamska optimizacija: Rad na algoritmima koji se primjenjuju u okviru programa za akviziciju i vizuelizaciju može poslužiti kao osnov za unapređenje performansi rada sa vremenskog aspekta. Takođe, efikasniji algoritmi mogu uticati na manje zauzeće memorije, što je kod rada sa mikrokontrolerima od naročite važnosti.
- Softversko preračunavanje: Neposredno prije rekonstrukcije originalnih signala, neophodno je izvršiti pravilno preračunavanje vrijednosti. Kako bi se taj postupak adekvatno obavio, potrebno je u obzir uzeti naponski ofset, uticaj jednosmjernje komponente, šum, kao i rezoluciju AD konvertora.
- Referentni naponi: Stroža kontrola referentnih napona imala bi pozitivan efekat na preciznost rada sistema.
- Specifikacije AD konvertora: Upotreba mikrokontrolera sa AD konvertorom koji ima veću rezoluciju impliciralo bi da sistem postane osjetljiviji na manje promjene, a samim tim pouzdaniji i precizniji.
- Zaštita od prenapona i šuma: U ove svrhe moguće je koristiti različite mjere zaštite kao što su RC filtri ili keramički kondenzatori.

ZAHVALNICA

Autor se zahvaljuje prof. dr Miodragu Forcanu i mr Goranu Vukoviću na izdvojenom vremenu, podršci i velikoj pomoći u toku pisanja ovog rada. Rad je proistekao iz razvojnih aktivnosti u vezi sa formiranjem modernih eksperimentalnih mjernih sistema u okviru predmeta „Električna mjerenja” na Elektrotehničkom fakultetu Univerziteta u Istočnom Sarajevu.

LITERATURA

- [1] N. Đukanović, P. Trifunović, „Primena sistema za akviziciju podataka, merenje i upravljanje pri laboratorijskom ispitivanju materijala”, Rudarsko-geološki fakultet, Beograd
- [2] A. Sharkovich Ismailov, Z. Botirovich Jo'rayev, „Study of Arduino microcontroller board”, Andijan Machine Building Institute, „Science and Education” Scientific Journal, vol. 3, issue 3
- [3] I. Susnea, M. Mitescu, „Microcontrollers in Practice”, MICROELECTR., Springer (India) PVT. LTD, vol. 18
- [4] Zvanični Arduino sajt, <https://www.arduino.cc/>
- [5] P. A. Sanglier, P. Lewandowski, F. Klumb, „First experience on Python development for survey software, advantages and drawbacks”
- [6] Zvanični Python sajt, <https://www.python.org/>
- [7] Dokumentacija PySerial biblioteke, <https://pyserial.readthedocs.io/en/latest/pyserial.html>
- [8] Dokumentacija Matplotlib biblioteke, <https://matplotlib.org/stable/index.html>
- [9] Dokumentacija NumPy biblioteke, <https://numpy.org/>
- [10] Dokumentacija Time biblioteke, <https://docs.python.org/3/library/time.html>
- [11] Dokumentacija Csv biblioteke, <https://docs.python.org/3/library/csv.html>

ABSTRACT

This paper examines the application of Python libraries in systems for the acquisition and visualization of voltage signals based on Arduino microcontroller platforms. An experimental measurement system was developed in which analog voltage signals are acquired using the Arduino Mega 2560 development board and transmitted to a personal computer via USB serial communication for further processing and visualization within the Python environment. The functionality and compatibility of selected Python libraries for serial communication, numerical data processing, and graphical representation of measurement signals were analyzed. Special attention was devoted to evaluating system performance in the context of three-phase voltage measurement, including waveform reconstruction and real-time visualization. The obtained results indicate that the proposed architecture represents a functional, flexible, and cost-effective solution suitable for educational purposes, laboratory measurements and small-scale research applications.

Acquisition and Visualization of Voltage Signals Using Arduino Microcontrollers and a PC Computer Based on the Application of Python Libraries

Đorđe Kovačević