

Automatizacija prikupljanja podataka u laboratoriju

Pakasin, Tin

Master's thesis / Diplomski rad

2022

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:591448>

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

Download date / Datum preuzimanja: **2024-12-19**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

Tin Pakasin

Automatizacija prikupljanja podataka u
laboratoriju

Diplomski rad

Zagreb, 2022.

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

INTEGRIRANI PREDDIPLOMSKI I DIPLOMSKI SVEUČILIŠNI STUDIJ
FIZIKA I INFORMATIKA

Tin Pakasin

Diplomski rad

**Automatizacija prikupljanja podataka u
laboratoriju**

Voditelj diplomskog rada: Dr.sc. Neven Šantić

Ocjena diplomskog rada: _____

Povjerenstvo: 1. _____

2. _____

3. _____

Datum polaganja: _____

Zagreb, 2022

Zahvaljujem svome mentoru Nevenu Šantiću na pruženom strpljenju, znanju i velikoj pomoći pri pisanju ovog diplomskog rada. Hvala svima na institutu i u laboratoriju koji su pomogli i naučili me nešto novo.

Hvala svim prijateljima i Društvu koji su sa mnom prolazili sve dobre i loše strane studiranja. Najveću zahvalu dugujem svojoj obitelji i roditeljima bez kojih ništa ovo ne bi bilo moguće.

Hvala vam svima na strpljenu i što ste uvijek bili uz mene i pružali mi podršku

Sažetak

U modernim laboratorijima u atomskoj fizici, podatci se kontinuirano prikupljaju i bilježe. Zbog utjecaja koji temperatura i drugi atmosferski uvjeti imaju na vrlo osjetljive eksperimente, na primjer, optičke atomske satove [1], nužno je mjeriti i bilježiti ove parametre kako bi se postigla rekordna točnost mjerenja vremena koji ovakvi satovi omogućuju. U ovom radu je opisana izrada sustava za automatizirano, kontinuirano mjerenje, bilježenje i interaktivno prikazivanje parametara u Laboratoriju za hladne atome na Institutu za fiziku. Mjerenja su bazirana na Labjack i Pico uređajima. Ostvareno je kontinuirano mjerenje i bilježenje temperature na više točaka u laboratoriju. Most između mjerenja na uređajima i spremanja podataka u bazu baziran je na programskom jeziku Python. Podaci će se spremati u InfluxDB bazu podataka, te se mogu prikazivati interaktivno i u stvarnom vremenu. Ovaj postav služit će za daljnju nadogradnju i priključivanje novih mjernih instrumenata čiji će se podaci moći obrađivati, spremati i nadograđivati kako bude potrebno za vrijeme trajanja eksperimenta.

Ključne riječi:

Automatizacija, Labjack, Pico, Python, baza podataka, , InfluxDB baza podataka

Automation of data acquisition in the laboratory

Abstract

In modern laboratories in atomic physics, data is continuously collected and logged. Due to the influence that temperature and other atmospheric conditions have on very sensitive experiments, for example, on optical atomic clocks, it is necessary to measure and log these parameters in order to achieve the record accuracy of time measurements that such clocks allow. This thesis describes the development of a system for automated, continuous measurement, logging and interactive display of parameters in the Laboratory for Cold Atoms at the Institute of Physics. Measurements are based on Labjack and Pico devices. Continuous measurement and logging of temperature at several points in the laboratory was achieved. The bridge between measuring on devices and saving data to a database is based on the Python programming language. The data will be stored in a InfluxDB database and can be displayed interactively and in real time. This setup will be used for further upgrading and connection of new measuring instruments whose data will be able to be processed, logged and upgraded as needed during the experiment.

Keywords:

Data automation, Labjack, Pico, Python, InfluxDB, database

Tin Pakasin

Sadržaj

1	Uvod	4
2	Python.....	6
3	Baza podataka.....	7
	3.1 Relacijske baze podataka	8
	3.2 Arhitektura relacijskih baza podataka.....	10
	3.3 Baza podataka vremenskog niza.....	11
4	Arhitektura.....	15
	4.1 Pico TC-08.....	16
	4.2 Labjack T7	20
5	Senzori.....	23
	5.1 Pt100 i Pt1000.....	23
	5.2 Termistori.....	25
	5.3 Steinhart-Hart jednadžba	26
6	Implementacija programskog okvira.....	27
	6.1 Razrada koda.....	28
7	Mjerenja i umjeravanje	33
	7.1 Grafovi	36
	7.2 InfluxDB	37
8	Metodički dio.....	41
9	Zaključak	48
10	Literatura	49

1 Uvod

U novom krilu Instituta za fiziku smješten je odsjek Instituta pod nazivom „Centar za napredne laserske tehnike“ u kojem će se provoditi brojni eksperimenti. Jedan od tih eksperimenata je i izrada atomskog optičkog sata baziranog na atomima stroncija. Kao i svi satovi, za izradu nam je potreban sustav s regularnim periodičkim ponavljanjima; odnosno sustav čiji se ciklus odvija na konstantnoj frekvenciji. Time možemo definirati stabilnu frekvencijsku referencu i osnovnu jedinicu vremena. Atomi se satovi, za razliku od klasičnih satova, temelje na oscilacijama elektrona oko atomske jezgre. Za izradu ovakvih i sličnih eksperimenata, bitno je imati preciznu i točnu mogućnost mjerenja što više varijabli koje mogu utjecati na eksperiment.

U slučaju optičkog sata to mogu biti temperatura i/ili tlak u prostoriji kojoj se nalaze. Kako bi skratili vrijeme potrebno za očitavanje mjerenja na nekoliko lokacija, željeli smo uspostaviti sustav koji bi mjerio i bilježio podatke umjesto čovjeka. Samim time smanjuje se vrijeme potrebno za očitavanje na različitim položajima u prostoriji, te mogućnost pogrešaka koje se mogu potkrasti u tom procesu.

U prostoriji je postavljeno računalo koje će služiti kao server i centralno mjesto za pohranu i obradu podataka. Na njega je spojeno nekoliko uređaja (specifično Pico TC-08 termopar i Labjack T7 automatizacijski uređaj), koji služe za prikupljanje i obradu podataka koji se zatim šalju na glavno računalo, te spremaju u bazu podataka koju pokreće glavno računalo. Serversko (glavno računalo) osigurava se od nedostatka opskrbe električnom energijom i ostalim mogućim problemima UPS (engl. *Uninterrupted power supply*) napajanjem. Takvo napajanje omogućava neometan način rada svih uređaja u laboratoriju u slučaju nedostatka struje i ostalih smetnji. Kako uređaji za mjerenje također dobivaju napajanje iz glavnog (serverskog) računala, time je pokriven i problem njihovog neometanog rada. Nažalost, tu možemo uočiti problem, ako se nešto dogodi računalu što je nepovezano s napajanjem gradske mreže (problemi s matičnom pločom, prekid kontakta, i sl.) vidimo da će i uređaji za prikupljanje podataka također ostati bez napajanja. Međutim, taj problem se ne može izbjeći ako su oni odvojeni i spojeni na zasebne izvore napona, jer uređaji bez računala koje obrađuje te podatke ne služe svrsi. Kako smo ograničeni brojem mogućih izvora i lokacijama mjerenja, ovakav pristup činio se najbolji za naš specifični slučaj.

Kao i većini visoko specijaliziranih ustanova, pa tako institutima i laboratorijima visoko specifični zahtjevi koji nemaju standardiziranog rješenja ponekad postaju veća briga nego korist. Automatizacije iziskuju početni ulog i trud kako bi se kasnije vrijeme potrebno za bilježenje i obradu podataka svelo na minimum. Potrebni je točno definirati problem koji se želi riješiti i ishod do kojeg se želi doći. Imajući to na umu potrebno je odabrati prikladne sustave i tehnologije koje omogućavaju dolazak do željenog ishoda. Takav slučaj je i ovdje moguće opaziti. Široka lepeza mogućnosti koju je moguće koristiti i dalje je potrebno manualno prilagoditi situacijama koje su jedinstvene za svaki laboratorij. U slučaju ovog diplomskog rada, specifičnosti na koje je trebalo paziti su:

- Prostorna raspodjela mjernih uređaja koji se spajaju na jedno centralno računalo
- Raspon temperatura za svaki od senzora
- Preciznost i točnost senzora (potrebno je imati što manji šum u mjerenju)
- Mogućnost nadograđivanja programskog rješenja po potrebi za više senzora, više različitih vrsta senzora i njihovih mjernih raspona
- Korištenje iste baze podataka za lakši pristup svim budućim tipovima podataka

Kao rješenje ovih problema odabran je tehnološki okvir Python-a i Influx Database sustav.

2 Python

Python programski jezik je odabran kao prirodno rješenje za ovakve tipove problema. Želimo prikupiti i obraditi podatke s nekoliko mjernih uređaja koji su povezani na isto glavno računalo. Uz to ne želimo gubiti mogućnost proširenja i dodavanja novih uređaja, ako se za time pokaže potreba.

Python je interpretacijski, interaktivni, objektno orijentirani programski jezik koji uključuje module, dinamičko pisanje, visoku razinu dinamičkih struktura podataka i klasa. Također podržava i više različitih programskih paradigmi izvan objektno orijentiranog programiranja. Njegovo sučelje dopušta korištenje velikog broja sistemskih naredbi i programskih biblioteka. Proširiv je na C/C++ programske jezike, te prenosiv; radit će na bilo kojem uređaju sa Linux, macOS i Windows sustavu. Putem Python-a također imamo pokriveno slučajeve za internetske protokole (HTTP, FTP, POP,IMAP i dr.) [1]

Python kao jednostavan, čitak programski jezik je odličan izbor za automatizaciju. U daljnjem slučaju možemo ga koristiti i za obradu velikih količina podataka. Ako se sustav nastavi razvijati i nadograđivati, nije nezamislivo da on može i poslužiti kao osnova za obradu velikih podataka uz pomoć jedne od mnogih programskih biblioteka kao npr.:

- NumPy
- Keras
- Pandas
- PyTorch
- Scikit
- TensorFlow

3 Baza podataka

Bazu podataka možemo predočiti kao elektronički način pohrane podataka, to jest sustav čija je svrha pohrana i prikaz podataka na zahtjev korisnika. Kako bi baza zapravo obavila svoju svrhu, moramo i definirati što su to podatci, kakve tipove podataka razlikujemo, i kako ih možemo predočiti u bazi. Model podataka možemo definirati kao skup pravila koja određuje kako sve može izgledati logička struktura podataka. Model čini osnovu za oblikovanje i implementiranje baze, a zatim podatci koji ulaze u bazu moraju biti logički organizirani u skladu s modelom sustava upravljanja bazom podataka. Baza predstavlja višu razinu rada s podacima u odnosu na programske jezike. Ta viša razina rada očituje se u tome što tehnologija baze podataka nastoji (i u velikoj mjeri uspijeva) ispuniti sljedeće ciljeve:

- Fizička nezavisnost podataka
- Logička nezavisnost podataka
- Fleksibilnost pristupa podacima
- Istovremeni pristup podacima
- Čuvanje integriteta baze
- Mogućnost oporavka nakon kvara
- Zaštita od neovlaštene uporabe
- Zadovoljavajuća brzina pristupa podacima
- Mogućnost podešavanja i kontrole

Baze možemo podijeliti na nekoliko načina, prvenstveno po tipu baze:

- Relacijski model – zasnovan na matematičkom pojmu relacije. Baza se sastoji od tablica
- Mrežni model – Baza je predočena mrežom koja se sastoji od čvorova i usmjerenih lukova

- Hijerarhijski model – specijalni slučaj mrežnog modela. Baza je predočena stablom (hijerarhijom) ili skupom stabala
- Objektni model – inspiriran objektno orijentiranim programskim jezicima. Baza se sastoji od trajno pohranjenih objekata

Samo relacijske baze podataka uspijevaju ispuniti sve gore navedene ciljeve. Jedan podtip relacijskih baza je vremenski ovisna baza podataka koja će biti temelj i pokretač automatizacije i prikupljanja podataka. [2]

3.1 Relacijske baze podataka

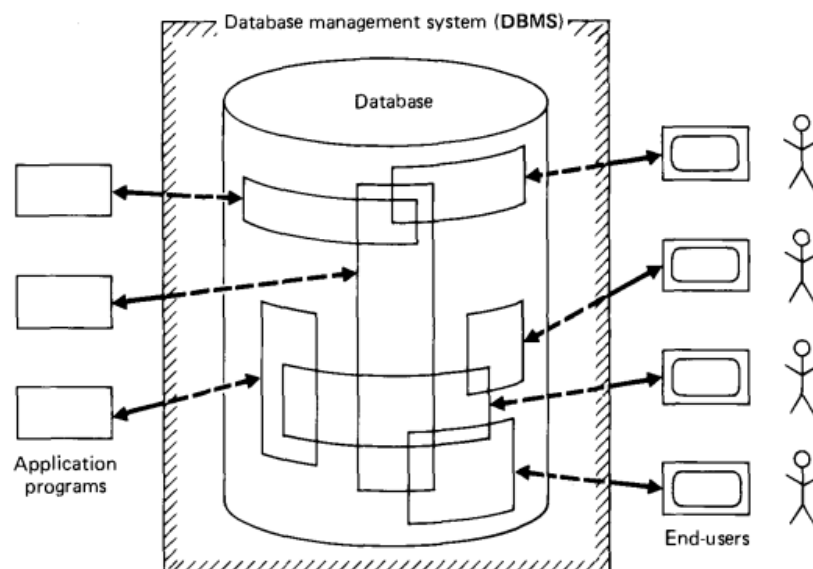
Da bi mogli objasniti što je vremenski ovisna baza podataka, i kako se ona razlikuje od relacijske, potrebno je prvo razjasniti što je relacijska baza podataka. Pojam relacijskih baza podataka pojavljuje se 1970.-ih u istraživačkom članku E.F.Codd u IBM-u [3]. Ona se sastoji od skupa povezanih tablica, odnosno relacija.

Raščlanimo ciljeve koji moraju biti zadovoljeni pri konstrukciji baze podataka.

- *Fizička nezavisnost podataka* - Razdvaja se logička definicija baze od njezine stvarne fizičke građe. To znači da, ako se fizička građa promjeni, (npr. podaci se prepisu u druge datoteke u drugim diskovima) to neće zahtijevati promjene u postojećim aplikacijama.
- *Logička nezavisnost podataka* - Razdvaja se globalna logička definicija cijele baze podataka od lokalne logičke jedinice za jednu aplikaciju. Znači, ako se globalna logička jedinica promjeni (npr. Uvede se novi zapis ili veza), to neće zahtijevati promjene na postojećim aplikacijama.
- *Fleksibilnost pristupa podacima* - U starijim mrežnim i hijerarhijskim bazama, načini pristupanja podacima bili su unaprijed definirani, dakle korisnik je mogao samo pretraživati podatke samo onim redom koji je previđen u vrijeme projektiranja i implementiranja baze. Danas se podrazumijeva da korisnik može slobodno prebirati po podacima i po svom nahođenju uspostavljati veze među podacima. Ovaj zahtjev je ključan i moguće ga je pronaći samo kod relacijskih baza podataka.

- *Istovremeni pristup podacima* - Baza mora omogućiti da se veći broj korisnika istovremeno koristi istim podacima
- *Zaštita od neovlaštenog korištenja* – Mora postojati pouzdana zaštita baze u slučaju kvara hardvera ili grešaka u radu sistemskog softvera
- *Zadovoljavajuća brzina pristupa* – Operacije s podacima moraju se odvijati brzo, u skladu s potrebama određene aplikacije. Na brzinu pristupa može se utjecati odabirom pogodnih fizičkih struktura podataka i izborom pogodnih algoritama za pretraživanje
- *Mogućnost podešavanja i kontrole* – Velika baza podataka zahtjeva stalnu brigu; praćenje performansi, mijenjanje parametara u fizičkoj građi, rutinsko pohranjivanje rezervnih kopija podataka, reguliranje ovlaštenja korisnika. Također, svrha baze se s vremenom može promijeniti, pa povremeno treba podesiti i logičku strukturu. [4]

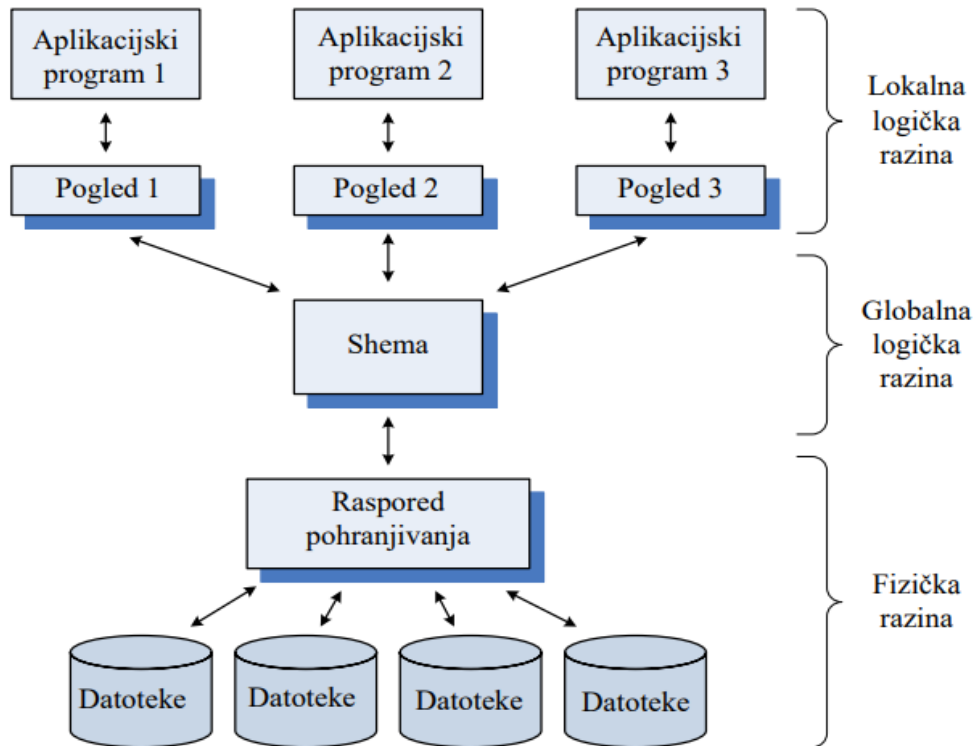
Rad na bazi podataka sastoji se nekoliko glavnih akcija: umetanje, brisanje, čitanje, izmjena. Ova 4 tipa akcija dovoljna su za sve mogućnosti rada nad podacima koji se nalaze u bazi. Umetanje se odnosi na dodavanje novog podatka u bazu. Brisanje se odnosi na uklanjanje podataka iz baze. Čitanje omogućava pregled već postojećih podataka u bazi, a izmjena dopušta mijenjanje nekih ili svih elemenata koji su već spremljeni u bazu.



Slika 1 – Apstraktna struktura baze podataka [2]

3.2 Arhitektura relacijskih baza podataka

Arhitektura baze podataka sastoji se od tri „sloja“ i sučelja među slojevima, kao što je prikazano na slici 2.



Slika 2 – arhitektura baze podataka [4]

Fizička razina odnosi se na fizički prikaz i raspored podataka na jedinicama vanjske memorije. Sama fizička razina može se podijeliti na više podrazina apstrakcije, od sasvim konkretnih staza i cilindara na disku, do apstraktnih pojmova datoteke i zapisa kakve susrećemo u klasičnim programskim jezicima. Raspored pohranjivanja opisuje kako se elementi logičke definicije baze preslikavaju na fizičke uređaje.

- Globalna logička razina odnose se na logičku strukturu cijele baze. Opis globalne logičke definicije naziva se *shema*. Shema je tekst ili dijagram koji definira logičku strukturu baze.
- Lokalna logička razina odnosi se na logičku predodžbu o dijelu baze kojim se koristi pojedina aplikacija. To je aspekt koji vidi korisnik ili aplikacijski programer. Opis jedne lokalne logičke definicije zove se *pogled*. To je tekst ili dijagram kojim se imenuju i definiraju svi lokalni tipovi podataka i veze među tim tipovima. [4]

Programi i korisnici fizičkoj bazi podataka ne pristupaju izravno, već dobivaju ili pohranjuju podatke posredstvom *Sustava za upravljanje bazom podataka*. Samo rješenje baze i pristup njoj u ovom radu, samostalno rješava program vremenski ovisne baze podataka pod nazivom *InfluxDB*.

U svrhu ovog rada fokusirat ćemo se na relacijski model i njegov specifični podtip, baza podataka vremenskog niza.

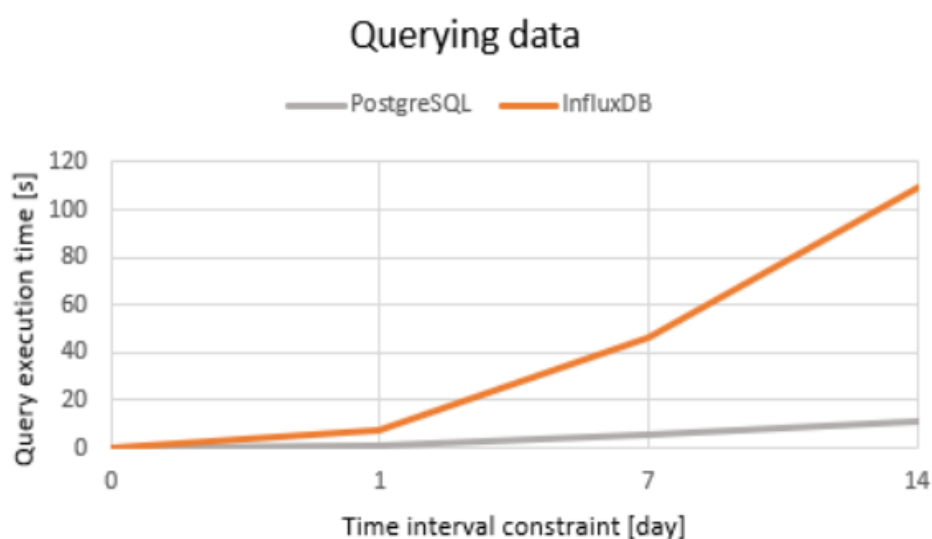
3.3 Baza podataka vremenskog niza

Jedna od tipova relacijskih baza podataka je i baza podataka vremenskog niza. Takva baza podataka je baza koja je optimizirana za vremenske nizove i podatke sa vremenskim žigom. Najčešće se koriste u radu s prikupljanjem podataka sa senzora, bankarskim sustavima, performansama aplikacije/programa, performanse serverskih uređaja i drugim sličnim podacima koji zahtijevaju obradu ili analitku. Takva baza je optimizirana za mjerenje promjena kroz vrijeme. Glavna i ključna razlika između relacijske baze i baze podataka vremenskog niza je u načinu kako se podaci spremaju i indeksiraju. Relacijske baze ne zahtijevaju poseban redoslijed dodavanja podataka u bazu, dok je u bazi podataka vremenskog niza vremenski žig taj koji je ključan za optimizaciju, jer je redoslijed umetanja u bazu automatski sortiran po vremenu umetanja. Taj način rada omogućuje optimiziranje određenih tipova zadataka koje izvršava baza podataka.

Pod agregacijom podataka smatramo transformaciju podataka prema nekom određenom parametru. Recimo, ako mjerenja obavljamo svakih 10 minuta, a prikazujemo graf u zadnjih 30 dana, to je previše točaka koje utječu na preglednost grafa. Ono što InfluxDB radi je, grupiranje svih mjernih točaka u danu i vraća srednju vrijednost. Ponavljajući tako proces za svaki dan, te na kraju imamo mjesečni pregled fluktuacija mjerenja (sa 30 točaka po danu). Prijašnja mjerenja se nisu izgubila, ona i dalje ostaju sačuvana te se njima može pristupiti

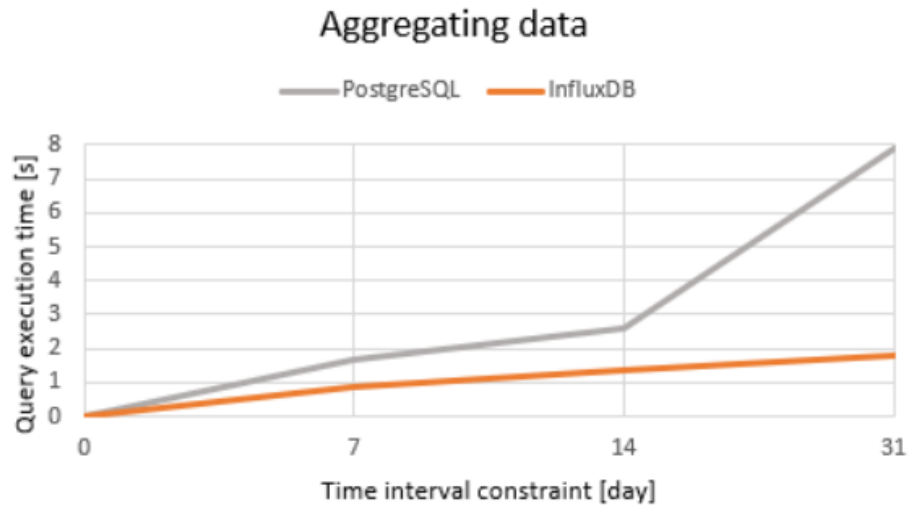
Vrijeme potrebno za agregaciju i rađanje osnovnih statističkih proračuna, grupiranje po indeksima ili atributima, osjetno je kraće u bazama podataka vremenskog niza. Naravno, te prednosti ne dolaze bez mana. InfluxDB ima 83.52% - 89.42% lošije performanse kada govorimo o pretraživanju neagregiranih podataka. [5]

Na slici 3 možemo vidjeti usporedbu pretraživanja podataka u PostgreSQL, jednoj od popularnih relacijskih baza podataka otvorenog koda i InfluxDB-a kada podaci nisu grupirani (svako mjerenje tretira se kao zaseban član). Uspoređujući upit u bazi na 3 različita vremenska perioda (1 dan, 7 dana i 14 dana), vidimo kako su tradicionalne baze podataka iznimno efikasne u tom pogledu, dok kod InfluxDB vidimo značajan porast u vremenu potrebnom za pretraživanje. S druge strane, možemo primjetiti kako je pristup agregiranju podataka kod InfluxDB znatno brži od standardnih relacijskih baza, te možemo ekstrapolirati da bi pri većem setu podataka ta razlika bila još i veća.



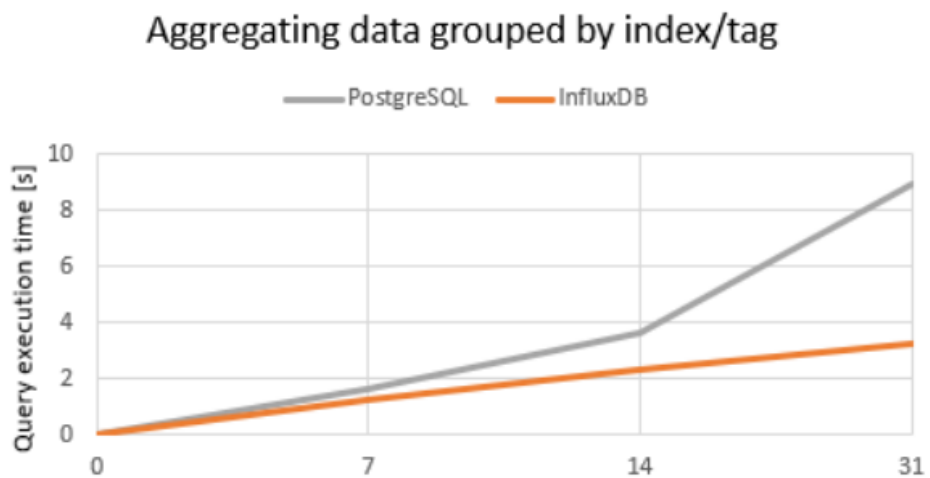
Slika 3 - Graf ovisnosti pretraživanja podataka; Usporedba PostgreSQL i InfluxDB

[5]

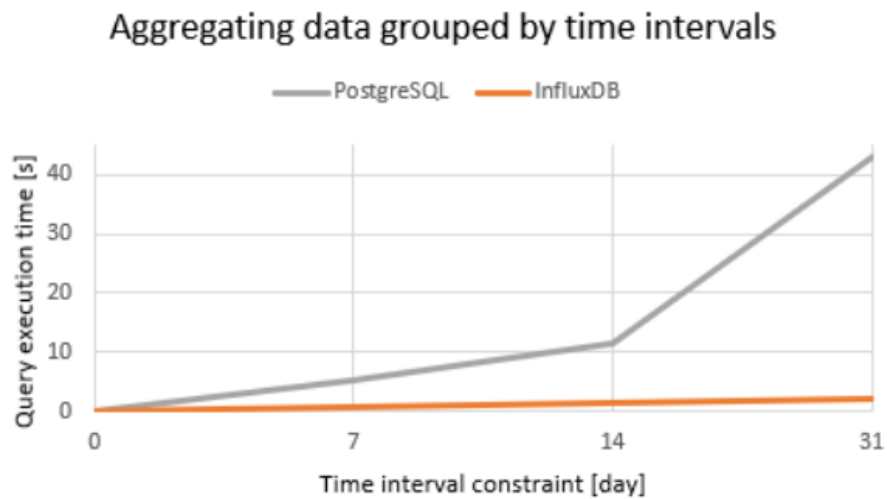


Slika 4 - Graf ovisnosti agregiranja podataka; Usporedba PostgreSQL i InfluxDB [5]

Međutim, kada isti postupak i usporedbu provodimo na podacima koji su agregirani i koje treba agregirati (pritom dodajemo i vremenski period od 31 dan kako bi usporedba bila vidljivija za veće skupove podataka), vidimo zašto baze podataka vremenskog niza imaju prednost. Njihovo vrijeme pretraživanja osjetno je brže (slike 5 i 6)



Slika 5 – Graf agregiranja podataka koji su grupirani pod zajedničkim indeksom [5]



Slika 6 – Graf grupiranja podataka ovisno o trajanju mjerenja [5]

Pri odlučivanju koju bazu podataka koristiti valja obratiti pozornost na to za što će se ti podaci koristiti. Možemo vidjeti da oba tipa baza imaju svoje prednosti i mane. InfluxDB osjetno je brži kada baratamo s podacima koji su agregirani i sekvencijalno uneseni u bazu, dok kod relacijskih baza primjećujemo da se na klasičnim tipovima podataka koji nisu vremenski ovisni ističu puno bolje performanse.

4 Arhitektura

Kako bi uspostavili automatsko prikupljanje podataka, potrebna su nam 3 tipa hardverske opreme:

1. Glavno računalo s bazom podataka
2. Pretvarač signala iz analognog u digitalni (*Analog-to-Digital Converter, AD pretvarač*)
3. Senzori

Uređaji koji su korišteni za prikupljanje i mjerenje podataka su Labjack T7 [6] i Pico TC-08 [7]. Na Pico TC-08 (slika 7) uređaju nalazi se 8 ulaznih kanala u koje se može spojiti 8 zasebnih senzora, svaki sa zasebnim kontrolama i mogućnostima mjerenja. Uređaj koristi 20-bitni AD pretvarač kako bi pretvorio napon generiran na termočlanku u digitalni zapis te iz tog napona odredio temperaturu. Napajanje uređaja vrši se putem USB-B kabela koji se spaja direktno u računalo. Time se i automatski rješava prijenos podataka s uređaja na računalo. Pošto želimo da računalo uvijek bude spremno primiti i pohranjivati podatke, a mjerni uređaj uvijek bude dostupan i spreman prikupljati ih, ovakvo rješenje daje stabilne rezultate.



Slika 7 – slika Pico TC-08 uređaja [8]

Labjack T7 (slika 8) je višenamjenski uređaj za prikupljanje podataka s velikim rasponom mogućnosti korištenja. U našem slučaju koristit ćemo ga za mjerenje temperature (kao i Pico TC-08). Njegovo spajanje slično je Pico uređaju, međutim Labjack T7 i T7 Pro verzije imaju opcije spajanja putem USB, Ethernet i WiFi-ja (samo T7 Pro verzija). Labjack u ovom slučaju ima više rezervnih opcija od Pico varijante. Primarni način spajanja je preko Ethernet, a ako mreža postane nedostupna, prijenos podataka i napajanje prebacuje se automatski na USB varijantu.



Slika 8 – Uređaj Labjack T7

4.1 Pico TC-08

Pico TC-08 je uređaj koji prikuplja podatke za spojene termoparove. U znanosti i tehnologiji termoparova, zbog svoje velike osjetljivosti imaju široku upotrebu kod preciznog mjerenja temperature, te ćemo dati osnovni princip rada termoparova. Najjednostavniji termopar čine dva povezana kontaktna spoja metala s različitim izlaznim radom. Prolazak struje kroz metal ne izaziva u njemu kemijske promjene. Električna vodljivost metala nije povezana s pomacima atoma u vodiču, već je određena gibanjem elektrona. Da bi se mogao objasniti elektronski karakter provođenja kroz metale, treba prihvatiti disocijaciju atoma na elektrone i pozitivne ione, zbog čega se u metalima nalazi veliki broj slobodnih elektrona. Ovi se elektroni mogu slobodno gibati po kristalnoj rešetci metala koju tvore ioni. Slobodni elektroni u metalima izvode kaotično termičko gibanje. Ako postoji i električno polje, ono usmjerava njihovo gibanje, čiji pomaci čine električnu struju.

Hipoteza o mogućnosti slobodnog gibanja elektrona u metalima može se potvrditi razmatranjem energije veza između elektrona i pozitivnih iona koji su postavljeni po vrhovima kristalne rešetke metala.

Uzmimo u razmatranje dva susjedna pozitivna iona A_1^+ i A_2^+ koji su na stalnoj udaljenosti. Svaki od njih se može smatrati točkastim nabojem. Potencijalna energija E_p uzajamnog djelovanja elektrona i pozitivnog iona iskazuje se izrazom:

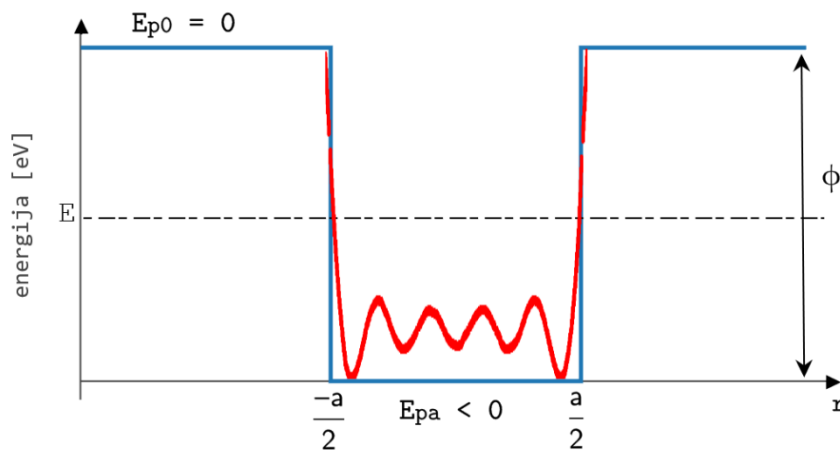
$$E_p = \frac{-konst}{r} \quad (1)$$

gdje je r udaljenost elektrona od iona.

Međutim, postoji više elektrona i atoma u kristalnoj rešetci. Međudjelovanje svih elektrona i atoma može se opisati zbrajanjem njihovih utjecaja na promatrani elektron (Slika 9). Suma svih doprinosa na grafu je prikazana crvenom bojom. Taj doprinos zatim aproksimiramo potencijalnom jamom (na grafu plava boja).

Potencijalna krivulja promatranog elektrona ima oblik duboke potencijalne jame. Područje u kojem je potencijalna energija konstantna, mnogo je veća od širine jame, i zato se može pretpostaviti da je potencijalna energija unutar metala konstantna i iznosi E_{pa} .

Pošto je potencijalna energija izvan metala E_{p0} veća od E_{pa} , elektron se unutar metala nalazi u potencijalnoj jami.



Slika 9 – Potencijalna energetska barijera metala koji se nalazi na intervalu $[-\frac{a}{2}, \frac{a}{2}]$

Ako se pretpostavi da je $E_{p0} = 0$, tada je $E_{pa} < 0$. Na slici 9 se vidi kako energija elektrona zadovoljava nejednakost $E_{p0} > E > E_{pa}$. Elektron se onda može gibati slobodno unutar metala, a iz njega može izaći jedino ako izvrši rad jednak $E_{p0} - E > 0$. Svaki metal ima svoj specifičan rad i on se naziva izlazni rad metala.

Kad se dva različita metala spoje, rad koji je potrebno izvršiti da elektron prijeđe iz jednog metala u drugi manji je od izlaznog rada. Elektroni koji imaju dovoljnu energiju

(prema statističkoj raspodjeli izvjestan broj elektrona i na nižim temperaturama ima neophodni minimum kinetičke energije da napusti metal) prelaze iz metala s nižim izlaznim radom u drugi metal sve dok se ovaj ne naelektrizira (negativno) do određenog potencijala. Na taj se način između dva različita spojena metala javlja potencijalna razlika, napon, koji se naziva kontaktni ili dodirni napon.

Veličina kontaktnog napona ovisi o vrsti vodiča koji se dodiruju, a mijenja se sa temperaturom. Kontaktni napon različitih parova metalnih vodiča je, općenito, mali i reda je veličine nekoliko mikro volta. Pojavu je otkrio Volta 1793. g., pa se naziva i Voltin efekt.

Ako se dva metala spoje, kroz njih ne teče struja, jer se razlike kontaktnih potencijala na mjestima spojeva A i B međusobno poništavaju. Ako se spoj A dovede na temperaturu različitu od temperature spoja B, pri čemu je $T_A > T_B$, kroz krug teče struja. Ukupna suma promjene kontaktnog potencijala tzv. elektromotorna sila (EMS), različita je od nule. Ona održava stalnu struju u krugu tzv. termostruju. Spoj metala koji na opisan način proizvodi termostruju naziva se termočlankom (termoparom).

Ova pojava naziva se i Seebeck efekt. Iako je cijeli niz eksperimenata pokazao da je elektromotorna sila termočlanka složena funkcija temperature, kao i da zavisi od prirode metala, postoje termočlanci, kod kojih je termoelektromotorna sila u dovoljno širokom intervalu proporcionalna razlici temperatura spojišta, odnosno $E = S(T_1 - T_2)$, gdje je S termoelektrični koeficijent para metala koji sačinjavaju termočlanak.

Termoelektrični koeficijent je brojno jednak elektromotornoj sili koja se javlja pri razlici temperatura spojeva od 1K. Stoga se termočlanci koriste za mjerenje temperature. To su u stvari električni termometri koji su vrlo osjetljivi i podešeni za mjerenje temperature u širokom rasponu vrijednosti, na primjer od temperature tekućeg helija (4.2 K) do sobne temperature (293 K) (termočlanak koji tvore: volfram-volfram+25% molibdena). Za ovakva mjerenja temperature u laboratorijima se koristi termočlanak kojeg čine dvije žice od različitih metala A i B. Pad napona se mjeri osjetljivim galvanometrom (u slučaju ovog rada, Pico TC-08 uređajem). Da bi se datim termočlankom temperatura mogla mjeriti, on se prethodno mora baždariti. Uz pretpostavku da posjedujemo baždarne termočlanke vrlo je lako realizirati razne uređaje za održavanje konstantne temperature

određenog procesa odnosno moguće je uz pomoć računala interaktivno voditi određeni tehnološki proces. [9]

Ovu je pojavu otkrio njemački fizičar Seebek 1821. god. Na tom efektu se temelji rad termoparova. Ovaj efekt često se i naziva Voltin efekt. Osnovno objašnjenje ove pojave glasi: Ako se duž vodiča javlja temperaturni gradijent na njegovim krajevima pojavit će se napon u skladu s relacijom:

$$dV_a = \alpha_a \frac{dT}{dx} dx \quad (2)$$

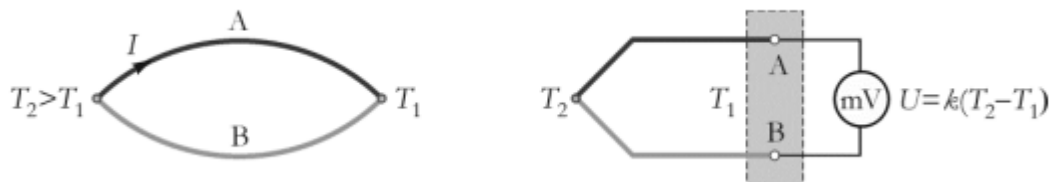
gdje je dT temperaturni gradijent po djeliću dužine dx , a α_a apsolutni Seebeckov koeficijent materijala. Ako je materijal homogen izraz se pojednostavnjuje:

$$dV = \alpha_a dT \quad (3)$$

Da bi dobili elektromotornu silu potrebno je sastaviti električni krug od dva različita materijala. Slobodni elektroni se tada u metalu ponašaju kao idealni plin, a u različitim materijalima energija i gustoća slobodnih elektrona nije ista. U trenutku kontakta između različitih materijala slobodni elektroni difundiraju kroz spoj pa se zbog različite koncentracije elektrona duž spoja javlja električno polje koje uspostavlja ravnotežu i difuzijski proces se zaustavlja. Ako su oba spoja na istoj temperaturi, efekti se poništavaju. Apsolutni Seebeckov koeficijent α_a je jedinstveno svojstvo materijala, a Seebeckov koeficijent k odnosi se na kombinaciju dvaju materijala:

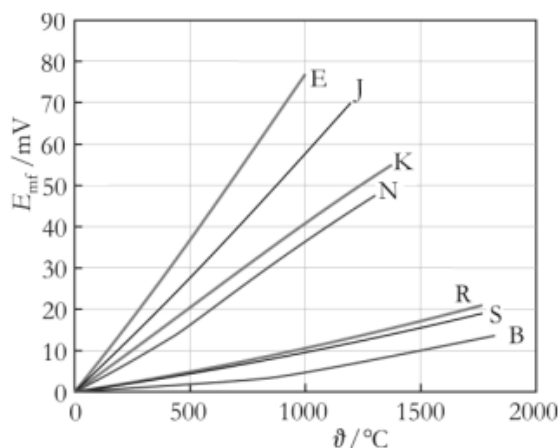
$$k = \alpha_a - \alpha_b \quad (4)$$

Na primjer, za a = željezo i b = konstantan Seebeckov koeficijent iznosi $k \sim 50 \mu\text{V}/^\circ\text{C}$. Struja koja teče petljom ovisi o više svojstava uključujući oblik i duljinu vodiča, dok je napon na krajevima vodiča isključivo funkcija materijala i razlike temperature.



Slika 10 – Seebeckov efekt u termoparovima [10]

Svaki par materijala pokazuje neka svojstva koja ga čine prikladnim za određenu primjenu. Njihova osjetljivost je različita, npr. veću osjetljivost ima par bakar-konstantan, a manju platina/rodij-platina. Statičke karakteristike nisu potpuno linearne, pa osjetljivost termoparova nije potpuno jednaka unutar cijelog mjernog područja (Slika 11). [10]



Slika 11 – Statičke značajke normiranih termoparova ovisno o tipu [10]

Karakteristike najpopularnijih termoparova su prikazani u tablici 1.

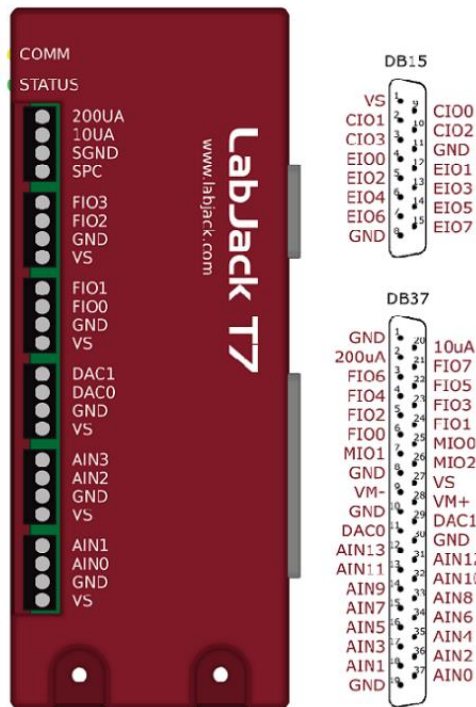
Tip	Materijal		Temperaturno Područje (°C)	U (Mv)
J	Željezo	Konstantan	- 200~ +750	- 7,89~+42,28
K	Chromel	Alumel	- 250~ +1350	- 6,404 ~ +54,125
E	Chromel	Konstantan	- 250~ +1000	- 9,719 ~ +76,358
T	Bakar	Konstantan	- 250~ +400	- 6,181 ~ +20,869
S	Platina	Pt + 10 %Rh	0 ~+1750	0 ~ +18,504
R	Platina	Pt + 13 %Rh	0 ~+1750	0 ~ +20,875
B	Pt + 6 % Rh	Pt + 30 %Rh	100~+1800	0,033~ +13,58
G	Volfram	W + 16 %Re	100~+2300	0,35 ~ +37,8

Tablica 1 – Mjerno područje normiranih termoparova [10]

4.2 Labjack T7

Labjack T7 je uređaj tvrtke Labjack i služi za digitalizaciju prikupljanja podataka. Uređaj nema tako široku dokumentaciju kao Pico TC-08, i zbog potpune digitalizacije i materijalnog vlasništva nije moguće doći do opširnog opisa unutarnjeg rada uređaja.

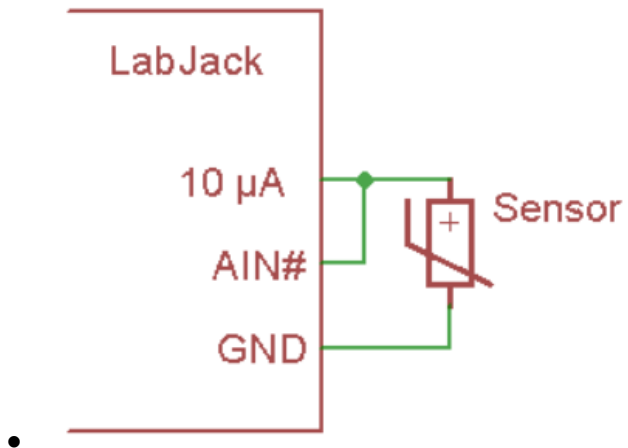
Međutim, mogućnost koju pruža je pristup svim podacima u registrima uređaja te mijenjanje istih što omogućava visoku razinu prilagodbe ovog uređaja potrebama korisnika.



Slika 12 – Svi ulazni priključci Labjack T7 [11]

Labjack T7 nudi sljedeće opcije [6]:

- 14 analognih ulaznih jedinica (analog-to-digital converter, AIN#) – 16-24 bitna, $\pm 10V$, $\pm 1V$, $\pm 0.1V$ i $\pm 0.01V$
- 7 fleksibilnih ulaz/izlaz (FIO)
- 23 digitalna ulaza
- 2 digital-to-analog konvertera (DAO) – rezolucije 12-bitna, raspon $\sim 0V$ to $\sim 5V$
- Uzemljenja (GND)
- Izvor napona (VS) od 0 do 5 Volta ($5V \pm 5\%$ regulirano)
- Izvor struje – $10\mu A$ i $200\mu A$

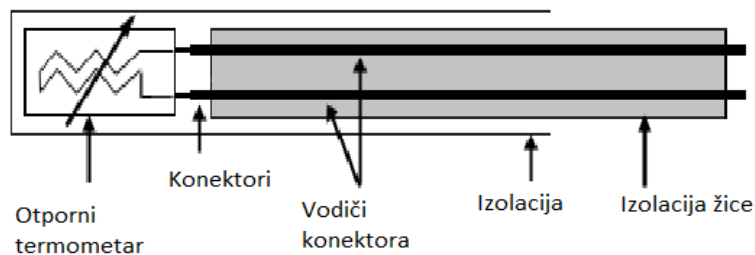


- Slika 13 – Shema korištenog spajanja senzora Pt100 na Labjack T7 [6]

5 Senzori

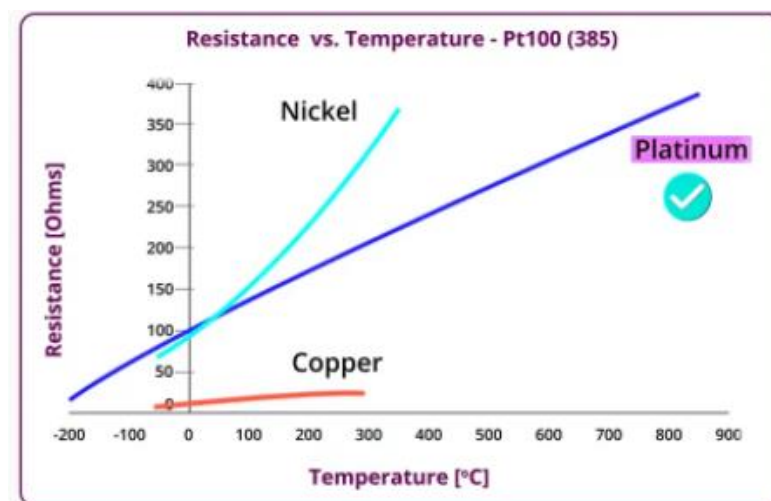
5.1 Pt100 i Pt1000

Senzori koje smo spajali na Labjack T7, u okviru ovog diplomskog rada, nazivaju se RTD senzori, odnosno otporno temperaturni detektori (Resistance Temperature Detectors). Namijenjeni su mjerenju temperature. Sastavljeni su od vodiča koji vodi do otpornog termometra. Vodič je obložen izolacijskom trakom, koja se najčešće proizvodi od PVC-a, silikonske gume ili nekih vrsta keramike. Cijeli sustav je još omotan vanjskom izolacijom od gume ili sličnih materijala (Slika 14). [12]



Slika 14 – Dijelovi RTD senzora [13]

Otporni termometar proizvodi od bakra, nikla, ili platine, ovisno o kojem temperaturum se planira mjeriti temperatura. U sve većem porastu je korištenje platine kao senzora zbog svojstva linearno rasta odnosa otpora i temperatura (Slika 15). Platina je kao element također vrlo inertan te ne reagira lagano s okolinom. Zbog svoje atomske mase 195,078 U, ima puno slobodnih elektrona koje mogu sudjelovati u transportu, te je time i dobar vodič električne struje.



Slika 15 Odnos otpora i temperature ovisno o materijalu [14]

Kada govorimo o RTD senzora, oni su napravljeni takvih dimenzija da na temperaturi od 0°C otpor iznosi točno 100 Ω ili 1000 Ω (otuda dolaze i nazivi Pt 100 ili Pt1000). Što ju čini ekstremno pogodnom za korištenje u izradi RTD senzora. Otpor u RTD senzora u ovisnosti u temperaturi određuje se pomoću izraza:

$$R_x = R_0(1 + \alpha T_x) \quad (5)$$

Gdje je R_x otpor izmjeren na nekoj temperaturi, R_0 iznos otpora na temperaturi 0°C, T_x temperatura na kojoj mjerimo otpor, a α je iznos promjene otpora po jedinici temperature. Za platinu on iznosi $\sim 0.385 \frac{\Omega}{^\circ\text{C}}$. Iznos α nije konstantan i mijenja se ovisno o čistoći korištene platine u senzoru. Kako bi povećali preciznost senzora, proizvođači uz senzor prilažu i standardiziranu tablicu otpora (tablica 2) iz koje možemo očitati standardizirane i umjerene vrijednosti na nekoj temperaturi što omogućava lakše korištenje i snalaženje.

Potrebno je samo naglasiti da, kada govorimo od RTD senzora, koristimo linearnu ovisnost temperature i otpora, što nije uvijek slučaj, ali za praktične primjene, to nam daje dovoljno dobru aproksimaciju i uvelike olakšava daljnji račun i korištenje uređaja i njegovih podataka.

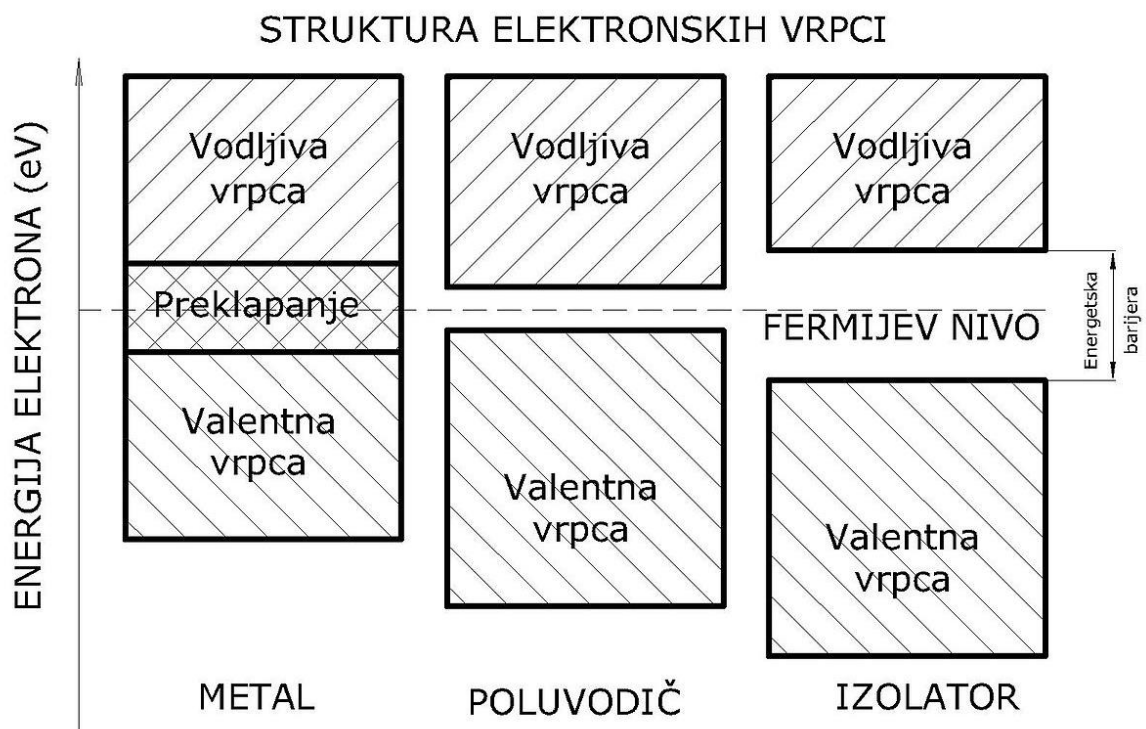
°C	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	°C
-60	76.33											-60
-50	80.31	79.91	79.51	79.11	78.72	78.32	77.92	77.52	77.12	76.73	76.33	-50
-40	84.27	83.87	83.48	83.08	82.69	82.29	81.89	81.50	81.10	80.70	80.31	-40
-30	88.22	87.83	87.43	87.04	86.64	86.25	85.85	85.46	85.06	84.67	84.27	-30
-20	92.16	91.77	91.37	90.98	90.59	90.19	89.80	89.40	89.01	88.62	88.22	-20
-10	96.09	95.69	95.30	94.91	94.52	94.12	93.73	93.34	92.95	92.55	92.1	-10
0	100.00	99.61	99.22	98.83	98.44	98.04	97.65	97.26	96.87	96.48	96.09	0
°C	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	°C
°C	0	1	2	3	4	5	6	7	8	9	10	°C
0	100.00	100.39	100.78	101.17	101.56	101.95	102.34	102.73	103.12	103.51	103.90	0
10	103.90	104.29	104.68	105.07	105.46	105.85	106.24	106.63	107.02	107.40	107.79	10
20	107.79	108.18	108.57	108.96	109.35	109.73	110.12	110.51	110.90	111.29	111.67	20
30	111.67	112.06	112.45	112.83	113.22	113.61	114.00	114.38	114.77	115.15	115.54	30
40	115.54	115.93	116.31	116.70	117.08	117.47	117.86	118.24	118.63	119.01	119.40	40
50	119.40	119.78	120.17	120.55	120.94	121.32	121.71	122.09	122.47	122.86	123.24	50
60	123.24	123.63	124.01	124.39	124.78	125.16	125.54	125.93	126.31	126.69	127.08	60
70	127.08	127.46	127.84	128.22	128.61	128.99	129.37	129.75	130.13	130.52	130.90	70
80	130.90	131.28	131.66	132.04	132.42	132.80	133.18	133.57	133.95	134.33	134.71	80
90	134.71	135.09	135.47	135.85	136.23	136.61	136.99	137.37	137.75	138.13	138.51	90
100	138.51	138.88	139.26	139.64	140.02	140.40	140.78	141.16	141.54	141.91	142.29	100
110	142.29	142.67	143.05	143.43	143.80	144.18	144.56	144.94	145.31	145.69	146.07	110
120	146.07	146.44	146.82	147.20	147.57	147.95	148.33	148.70	149.08	149.46	149.83	120
130	149.83	150.21	150.58	150.96	151.33	151.71	152.08	152.46	152.83	153.21	153.58	130
140	153.58	153.96	154.33	154.71	155.08	155.46	155.83	156.20	156.58	156.95	157.33	140
150	157.33	157.70	158.07	158.45	158.82	159.19	159.56	159.94	160.31	160.68	161.05	150
160	161.05											160
°C	0	1	2	3	4	5	6	7	8	9	10	°C

Tablica 2 – Odnosi otpora i temperatura RTD senzora [14]

5.2 Termistori

Termistori su električni termootpornici zasnovani na poluvodičkom materijalu. Električni otpor mu se znatno mijenja s promjenom temperature. Razlikujemo dvije vrste termistora: [15]

- PTC termistori (engl. *Positive Temperature Coefficient*) imaju pozitivni koeficijent temperaturene promjene otpora, to jest kako se temperatura povećava tako se povećava i električni otpor; temperaturno područje rada im je od $-50\text{ }^{\circ}\text{C}$ do $220\text{ }^{\circ}\text{C}$.
- NTC termistori (engl. *Negative Temperature Coefficient*) imaju negativni koeficijent temperaturene promjene otpora, to jest kako se temperatura povećava, tako se smanjuje električni otpor. Imaju nešto niže temperaturno područje rada od $-50\text{ }^{\circ}\text{C}$ do $150\text{ }^{\circ}\text{C}$.



Slika 16 – Struktura energetskih vrpca materijal [16]

Razlika NTC i PTC termistora dolazi iz materija od kojeg su građeni. Glavni doprinos otpora dolazi iz nečistoća, tj. nepravilnosti materijala (statički defekti) i od fonona koji

predstavljaju vremenski promjenjiva odstupanja od idealne kristalne rešetke zbog gibanja naboja (dinamički defekti).

Negativni temperaturni koeficijent primarno je vidljiv kod poluvodiča. Dolazi iz činjenice da povećanjem temperature više elektrona ima energiju da „preskoči“ zabranjeni pojas i prijeđe iz valentne u vodljivu vrpcu. Samim time oni doprinose povećanju vodljivosti (a time i smanjenju otpora) jer jednostavno postoji više naboja koji može voditi struju.

S druge strane, pozitivni temperaturni koeficijent, može se primijetiti kod metala (npr. platine) i dielektrika (izolatora). Povećanje temperature dovodi do jačeg titranja kristalne rešetke, te dolazi do češćih sudara elektrona sa atomima kristalne rešetke. Ti sudari smanjuju driftnu brzinu elektrona što dovodi do smanjenja struje, odnosno povećanja otpora.

5.3 Steinhart-Hart jednadžba

Steinhart-hard jednadžba je empirijski izvedena jednadžba koja predstavlja model otpornosti poluvodiča na različitim temperaturama. Izraz glasi:

$$\frac{1}{T} = A + B \ln \frac{R}{R_{25}} + C \left(\ln \frac{R}{R_{25}} \right)^2 + D \left(\ln \frac{R}{R_{25}} \right)^3 \quad (6)$$

gdje je: T temperatura u kelvinima,

R_{25} otpor na temperaturi 25°C,

R otpor na traženoj temperaturi,

A, B i C su Steinhart-Hart koeficijenti.

Steinhart-Hart koeficijenti ovise o modelu termistora i temperaturnom intervalu u kojem se koristi. Ona se koristi pri izračunu temperature na Labjack uređaju. Pt100 i Pt1000 senzori se spajaju na Labjack-ov izvor struje od 200 μ A. Kroz vodič i senzor se pušta struja. Senzor Pt100 ima otpor 100 Ω , a Pt1000 ima otpor 1000 Ω na temperaturi od 0 °C. Pad napona na senzoru automatski očitava Labjack, i preko Steinhart-Hart jednadžbe računa temperaturu. Pad napona može se izračunati prema Ohmovom zakonu, gdje su temperaturno ovisni otpori prikazani u tablici 2. Pad otpora se se automatski pretvara u temperaturu na samom uređaju, te se zatim šalje na računalo gdje se sprema u bazu podataka.

6 Implementacija programskog okvira

U prijašnjem djelu teksta opisana je struktura, načina rada i elementi koji su se koristili u postavu ovog eksperimenta. O ovom poglavlju bit će opisan postav i kod koji je korišten za automatizaciju i mjerenja.

Započeli smo sa spajanjem Pico TC-08 i Labjack T7 uređaja na računalo. Cilj u ovom koraku je uspostaviti vezu između tih uređaja i Python programskoj jezika. Koraci koje je potrebno pratiti su:

Instalacija potrebnih upravljačkih programa za Pico uređaj na Linux operativnom sustavu pomoću komande:

```
sudo apt-get install libusbtc08
```

Libusbtc08 omogućava kontrolu i komunikaciju između uređaja i računala. U njemu se nalaze funkcije za očitavanje i kontinuirani prijenos podataka kao i općenite funkcije za kontrolu i održavanje konekcije. Sve funkcije pisane su u C/C++ programskom jeziku.

Kako bi u Pythonu mogli očitati C/C++ funkcije potrebna je i instalacija programskog paketa *ctypes*. *Ctypes* služe kao svojevrsni omotač oko C/C++ koda koji omogućava pozivanje funkcija u Pythonu. Omotač prevodi C/C++ kod u kod koji Python može interpretirati i pokrenuti. Koristeći SWIG (Simple Wrapper Interface Generator), proces je zapravo jednostavan i radi „out-of-the-box“.

Pokreće se jednostavna komanda nad datotekom *usbtc08.i* u *libusbtc08* direktoriju.

```
swig -python usbtc08.i
```

Napomena: *Ctypes* dolazi uz nove verzije Python (2.5+), međutim ako se koristi Python ispod verzije 2.5, moguće da ih je potrebno manualno instalirati putem komandne linije. U komandnu liniju potrebno je unijeti:

```
pip install ctypes
```

Instalacija PicoSDK izvodi se pomoću jednostavne komande:

```
pip install picosdk
```

Koristeći neki od programskih sučelja, sada možemo započeti prikupljanje podataka.

Kako bi započeli prikupljanje podataka, potrebno je uspostaviti vezu Pythona i Pico uređaja. Za potrebe ovog diplomskog rada, želja je bila automatizirati prikupljanje podataka o temperaturi u prostorijama laboratorija.

6.1 Razrada koda

Prije početka koda imamo uvoz potrebnih biblioteka za rad kao što je ranije objašnjeno. *Ctypes, libusbtc08 i picoSDK*.

Za početak, uspostavljamo vezu s Pico uređajem uz pomoć `chandle` objekta. Objekt se razlikuje od `handle` objekta u jednom svojstvu, a to je da je `chandle` automatski izbrisan nakon završetka svoga rada. [17]. Kako bi lakše objasnili ovaj dio, možemo iskoristiti analogiju rukovanja. Uspostavlja se rukovanje (*handshake*) između uređaja i računala. Definiramo tip varijable, kao što je inače praksa u C/C++ jezicima, kao 16-bitni integer. Za to koristimo funkcije definirane u PicoSDK biblioteci (računalni kod 1).

```
chandle = ctypes.c_int16()
status = {}
data = []

#open unit
status["open_unit"] = tc08.usb_tc08_open_unit()
assert_pico2000_ok(status["open_unit"])
chandle = status["open_unit"]
```

Računalni kod 1 – Inicijalizacija uređaja

Postavljamo unaprijed ugrađeni filter digitalnog signala na 50Hz kako bi smanjili interferenciju od signala gradske mreže (računalni kod 2). U funkciju se prosljeđuje 0 koja odgovara iznosu od 50 Hz.

```
status["set_mains"] = tc08.usb_tc08_set_mains(chandle,0)
assert_pico2000_ok(status["set_mains"])
```

Računalni kod 2 – Smanjenje interferencije gradske mreže

Zatim želimo inicijalizirati sve kanale koji će biti korišteni pri mjerenju. U trenutnom postavu samo nekoliko kanala je bilo iskorišteno, ali svih 8 je inicijalizirano. Razlog tomu je što smo željeli olakšati prebacivanje i ubacivanje novih uređaja u daljnjem radu. Inicijalizacija svih 8 kanala pruža opciju ukopčavanja novog uređaja dok je program upaljen bez potrebe za gašenjem i novim pokretanjem programa. Postavlja se i vrijednost 75 kao indikator koji termoelement se koriste pri očitavanju temperature. U našem slučaju je

to Tip K (računalni kod 3). To je legura nikla i kromela (koji se pak sastoji od 90% nikla i 10% kroma).

```
typeK = ctypes.c_int8(75)

error_flag = 0
try:
    for k in range (1,9):
        status["set_channel"] = tc08.usb_tc08_set_channel(chandle, k, typeK)
        assert_pico2000_ok(status["set_channel"])
```

Računalni kod 3 – Postav mjernih kanala Pico uređaja

Kada smo uspostavili vezu s uređajem i dali sve potrebne varijable, sada možemo započeti očitavanje. Pošto želimo kontinuirano mjerenja bez pokretanja programa, sva mjerenja odrađujemo u beskonačnoj *While True* petlji (računalni kod 4). *Minimum sampling interval* je najmanji mogući interval između kojih je moguće mjeriti podatke. On u prosjeku iznosi 100 ms po aktivnom kanalu i njegovo izvođenje odrađeno je u PicoSDK-u.

```
while True:
    status["get_minimum_interval_ms"] = tc08.usb_tc08_get_minimum_interval_ms(chandle)
    assert_pico2000_ok(status["get_minimum_interval_ms"])

    temp = (ctypes.c_float * 9)()
    overflow = ctypes.c_int16(0)
    units = tc08.USBTC08_UNITS["USBTC08_UNITS_CENTIGRADE"]
    status["get_single"] = tc08.usb_tc08_get_single(chandle, ctypes.byref(temp), types.byref(overflow), units)
    assert_pico2000_ok(status["get_single"])

    i=0
    for elem in temp:
        if i==0:
            i+=1
            ch_name="Cold Junction"
            dbWrite(bucket,org,ch_name)
        else:
            ch_name = "Channel " + str(i)
            dbWrite(bucket,org,ch_name)
            i+=1

    #Wait time between readings in seconds, 60 = 1 min, 300 = 5 min, 600 = 10 min
    time.sleep(5)
```

Računalni kod 4 - Očitavanje sa Pico uređaja

Ovdje možemo primijetiti funkciju *dbWrite* koja zapisuje podatke u bazu podataka. O njoj će biti više riječi u idućem poglavlju. *Status[„get single“]* lista je ona koja odrađuje mjerenje. Preko nje se dohvaćaju podaci koji dolaze iz uređaja. U nju se prosljeđuje

direktan zapis vrijednosti koju mjerimo. Alternativni zapis može se izvesti uz pomoć *byval* funkcije koja šalje kopiju originalnog zapisa. Takav način ekvivalentan je kreiranju nove varijable s istim iznosom i prosljeđivanje te nove varijable u zapis. Kod osjetljivih i bitnih podataka, kako bi se izbjeglo neželjeno miješanje varijabli, preporuča se korištenje *byref* funkcije.

```
except:
    if error_flag == 1 :
        print("Pico closed forcefully. Re-run the program")
        picoClose()
    if error_flag == 0 :
        picoClose()
else:
    picoClose()
```

Računalni kod 5 - *try-except-else* metoda grešaka

Vješto oko moglo je uočiti i *try-except-else* metodu korekcije grešaka. Ako nešto pođe po zlu, želimo izbaciti program iz funkcije kako bi se spriječilo upisivanje loših i beskorisnih podataka u bazu. *Error_flag* varijabla provjerava je li program doživio pogrešku ili je prisilno ugašen korisnikovom željom. Treba naglasiti kako nije dovoljno samo iskopčati uređaj iz računala jer se ne zatvara način komuniciranja između uređaja i računala na pravilan način. Može doći do situacije gdje računalo i dalje čeka podatke od uređaja koji više nije spojen na njega, te se takva situacija nebrojeno puta dogodila. Ovakvim pristupom osiguravamo se da ne moramo gasiti i paliti računalo prilikom svake pogreške u radu.

Labjack T7 ima puno lakši način spajanja od Pico uređaja. Potrebno je pročitati korisnički priručnik (kako bi se odredio željeni način korištenja [18]). U ovom slučaju, koristit će se RTD temperaturni senzori (specifično Pt100 i Pt1000).

Instalacija se svodi na preuzimanje LJM biblioteke za korištenje i sa službene Labjack stranice [19]. Potrebno je preuzeti i instalirati sve potrebne programe, kao i u Pico slučaju. Softver potreban za to nalazi se na [20].

Iz direktorija u kojem se nalazi preuzeta datoteka, potrebno je pokrenuti sljedeće naredbe:

```
python -m pip install labjack-ljm
python setup.py install
```

Nakon inicijalizacije biblioteka, nije bilo većih problema pri postavljanju ostatka programa.

```
from labjack import ljm
import time
import influxdb_client

handle = ljm.openS("ANY", "ANY", "ANY")
```

Računalni kod 6 – Umetanje biblioteka LabJack Manager

Započinjemo s otvaranjem i uspostavljanjem veze sa Labjackom. LabJackManager (LJM) sustav rješava sve probleme bez dodatnih napora.

```
def steinhartHeart_equation():
    nameArray = ["AINO_EF_INDEX",
                 "AINO_EF_CONFIG_A",
                 "AINO_EF_CONFIG_B",
                 "AINO_EF_CONFIG_F",
                 "AINO_EF_CONFIG_G",
                 "AINO_EF_CONFIG_H",
                 "AINO_EF_CONFIG_I",
                 "AINO_EF_CONFIG_J"]

    valueArray = [50,1,0,1,1.128080355e-3,2.342890601e-4,0,0.8710181288e-7 ]

    numFrames = len(nameArray)
    ljm.eWriteNames(handle,numFrames,nameArray,valueArray)
```

Računalni kod 7 – Funkcija za račun temperature pomoću Steinhart-Hart jednadžbe

Definiramo funkciju koja će preko Steinhart-Hart jednadžbe računati temperaturu na termistoru (računalni kod 7). AIN je oznaka ulaznog kanala koji se koristi pri spajanju. U ovom slučaju je to kanal 0. Opcije EF_CONFIG služe za namještanje izračuna ovisno o namjeni. U našem slučaju je to NTC termistor. Registri navedeni u funkciji su redom:

- EF_CONFIG_A – služi za odabir temperaturne skale (0 za Kelvine, 1 za stupanj Celzijus, 2 za stupanj Fahrenheit)
- EF_CONFIG_B – služi za odabir načina spajanja senzora u uređaj
- EF_CONFIG_F – upisuje se nominalna vrijednost otpora termistora na 25 °C
- EF_CONFIG_G – koeficijent A iz Steinhart-Hart jednadžbe (6)
- EF_CONFIG_H – koeficijent B iz Steinhart-Hart jednadžbe (6)
- EF_CONFIG_I – koeficijent C iz Steinhart-Hart jednadžbe (6)
- EF_CONFIG_J – koeficijent D iz Steinhart-Hart jednadžbe (6)

Ponekad se član $D \left(\ln \frac{R}{R_{25}} \right)^3$ iz jednačbe (6) izbacuje, te se on može postaviti na nulu upisom u registar EF_CONFIG_J= 0.

Podaci se upisuju u registar prilikom svakog očitavanja putem *ljm.eWriteNames* funkcije koja kao parametre redom prima:

- labjack koji je trenutno aktivan (*handle*),
- broj podataka koje ćemo upisivati (*numFrames*),
- registre kojima pristupamo (*numArrays*), te
- vrijednosti koji se upisuju u registre.

Očitavanje se izvodi jednostavnim čitanjem stanja registra na uređaju u koji se upisuju rezultati mjerenja. On pohranjuje taj rezultat do trenutka kada novo mjerenje bude pokrenuto i novi rezultat tada zauzima taj registar.

```
elem = ljm.eReadName(handle, "AINO_EF_READ_A")
```

Računalni kod 8 – Očitavanje podataka iz LabJack uređaja

7 Mjerenja i umjeravanje

Kako bi se uvjerali u ispravnost izmjerenih podataka, konstruirali smo mali eksperiment za provjeru točnosti senzora. U čisti spremnik dodali smo zamrznutu destiliranu vodu, te provjeravali kakva mjerenja dobivamo od 3 senzora (RTD Pt100, Pt1000 i Th10k termistor).

Mješavina vode i leda ostaje na 0 stupnjeva zbog fizikalne veličine koja se naziva latentna toplina taljenja. Taljenje je proces prelaska iz čvrstog u tekuće agregatno stanje.

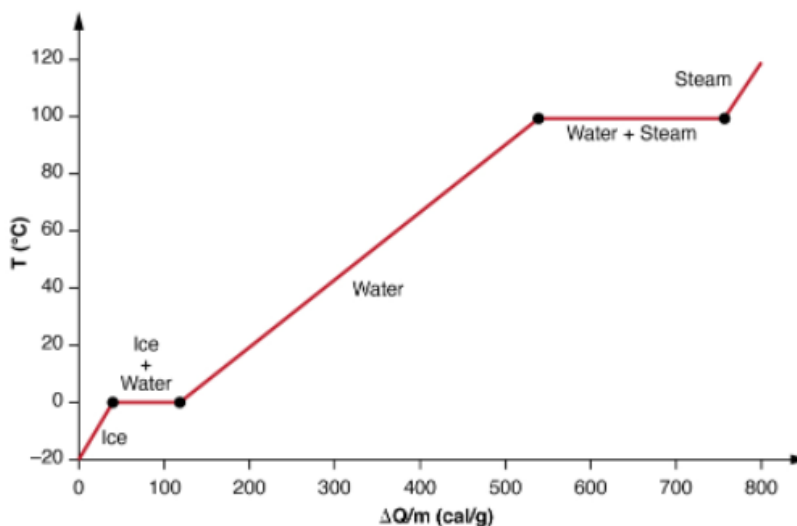
Latentnu toplinu možemo zamisliti kao energiju potrebnu da bi se savladale međumolekulske interakcije. Idealna kristalna rešetka održava svoju strukturu silama privlačenja između idealnih atoma, koji osciliraju oko ravnotežnog položaja. Povećanjem temperature, te oscilacije imaju veće amplitude. Kada dođu do kritične točke gdje je energija oscilirajućih atoma prevelika, oni se odvajaju od kristalne rešetke te prelaze u drugo agregatno stanje. Energija koja je potrebna da bi se dogodio taj prijelaz nazivamo latentna toplina taljenja λ [21].

Definirana je relacijom:

$$\lambda = \frac{Q}{m} \quad (7)$$

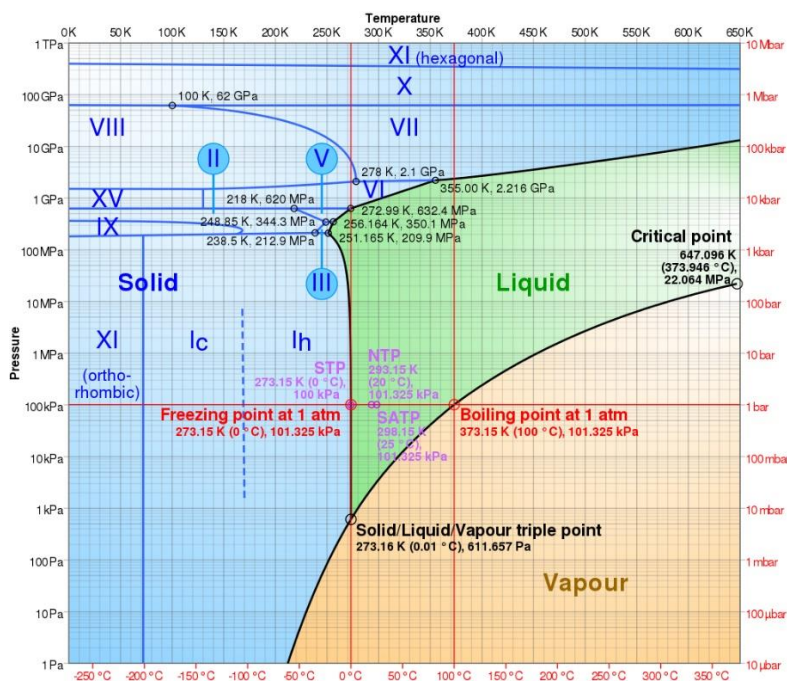
Gdje je:

- λ - latentna toplina
- Q - količina topline koju neka masa tvari mora predati ili primiti iz okoline kako bi promijenila agregatno stanje
- m – masa tvari



Slika 17 - Graf latentne topline za vodu [22]

Dakle možemo reći da latentna toplina „sprječava“ povećanje temperature dok traje fazni prijelaz, u ovom slučaju iz krutog u tekuće agregatno stanje.



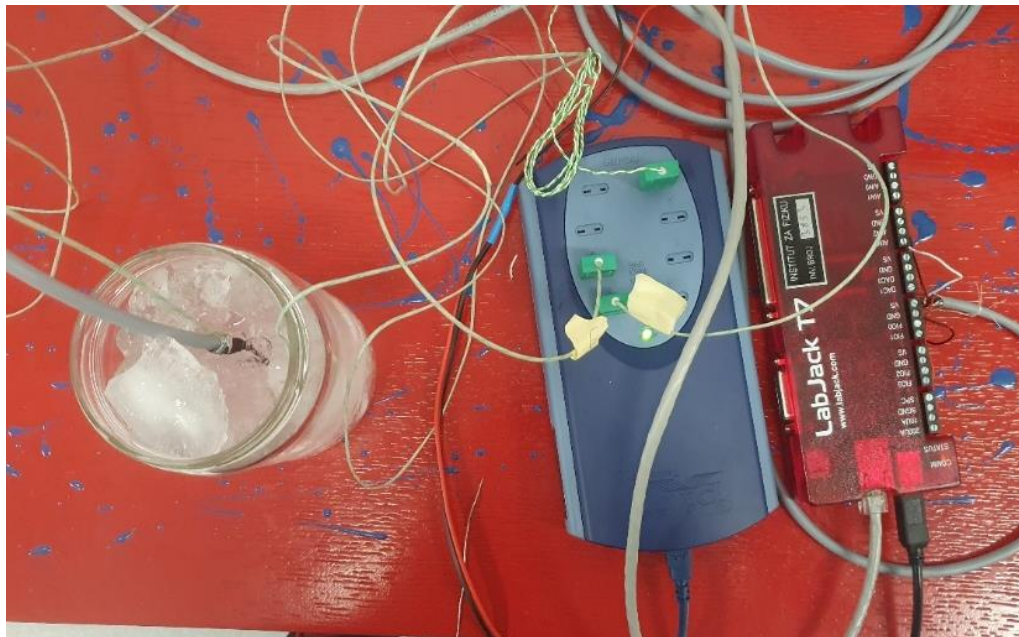
Slika 18 – Fazni dijagram vode [23]

Kako bi smanjili izmjenu topline sa okolinom, kartonsku kutiju smo obložili stiroporom, te u nju stavili posudu s ledenom kupkom. Mjerenja su obavljena jednom u

minutu s ukupnom dužinom trajanja od 90 minuta. Odabrane su točke koje ne prikazuju sobnu temperaturu kako bi graf bili čitljiviji.



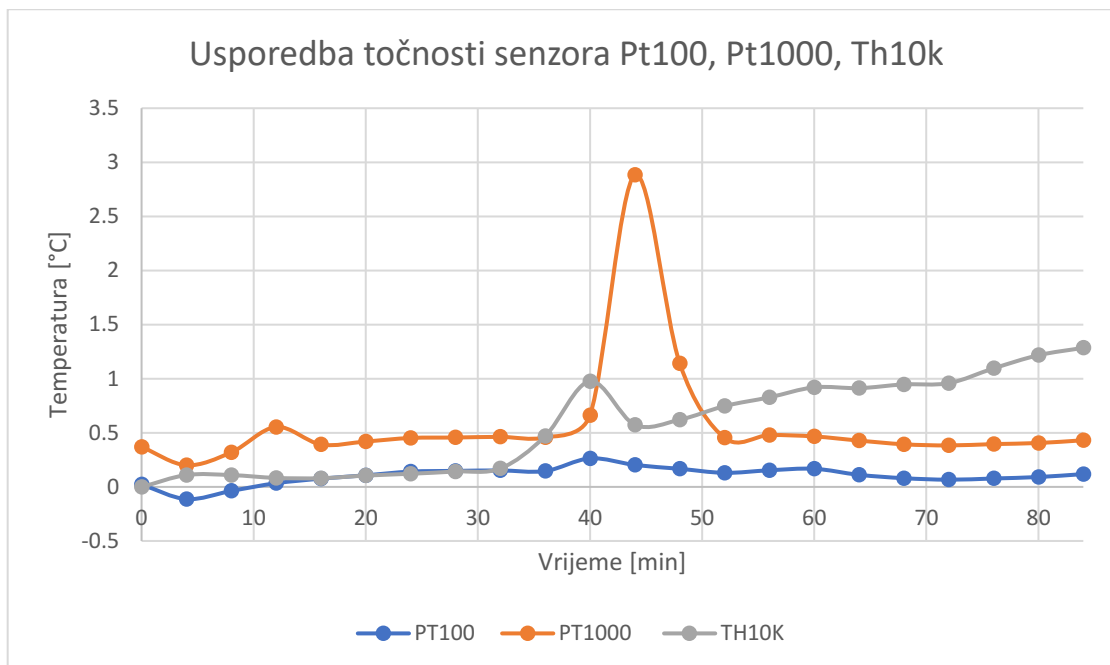
Slika 19 – Ledena kupka sa sensorima



Slika 20 – Labjack i Pico s ledenom kupkom

7.1 Grafovi

Rezultati koje dobivamo prikazani su na slici 21.



Slika 21 – Usporedba točnosti senzora Pt100, Pt1000 i Th10k

Na grafu možemo primijetiti skok između 40. i 50. minute za sva tri senzora. Možemo pretpostaviti da je došlo do naglog prijenosa topline, što ćemo pripisati pomicanju ili doticanju ledene kupke u kojoj se odvija mjerenje. Vidimo da su sva tri senzora brzo reagirala na to.

Usporedimo li mjerenja možemo zaključiti da:

1. Pt100 senzori najčešće imaju preciznost od ± 0.1 °C [24] Iz grafa vidimo da da mjerenja točno prikazuju takva odstupanja. Sva mjerenja u prvih 20 minuta nalaze se unutar preciznosti koja je propisana od strane proizvođača.
2. Preciznost Pt1000 senzora ovisi od klase izrade. Klasa A ima preciznost od ± 0.15 °C. Klasa B, koja je korištena u ovom postavu ima preciznost ± 0.40 °C. [25] Možemo primijetiti kako mjerenja se nalaze na oko 0.3 °C, te da su vrijednosti u skladu sa točnosti proizvođača.
3. Th10k thermistor ima najmanju točnost od 3 navedena senzora i ona iznosi ± 1 °C [26], što je i vidljivo da iz grafa. Mjerena temperatura raste s vremenom, dok taj isti

efekt ne vidimo na Pt100 i Pt1000 senzorima. To opažanje možemo pripisati grijanju vodiča i senzora radi struje koja teče kroz njega.

7.2 InfluxDB

Postav InfluxDB baze podataka izvodi se na sljedeći način. S web stranice InfluxDB [27] potrebno je preuzeti direktorij s najnovijom verzijom. U početku ovog rada instalirana je verzija 1.8, a do završetka rada izašla je verzija 2.0. Nadogradnje dolaze jako često (otprilike jednom mjesečno), te valja paziti na to žele li se novije verzije ažurno nadograđivati.

Preuzimanje direktorija započinje se naredbom:

```
wget https://dl.influxdata.com/influxdb/releases/influxdb2-2.2.0-xxx.deb
```

Nakon preuzimanja potrebno je odraditi instalaciju.

Napomena: Primijetiti da verzija nije dana ovdje, te je komandu potrebno prilagoditi najnovijoj verziji.

```
sudo dpkg -i influxdb2-2.2.0-xxx.deb
```

Nakon instalacije, potrebno je ponovno pokrenuti sistem i pokrenuti komandu koja će inicijalizirati bazu.

```
sudo service influxdb status
```

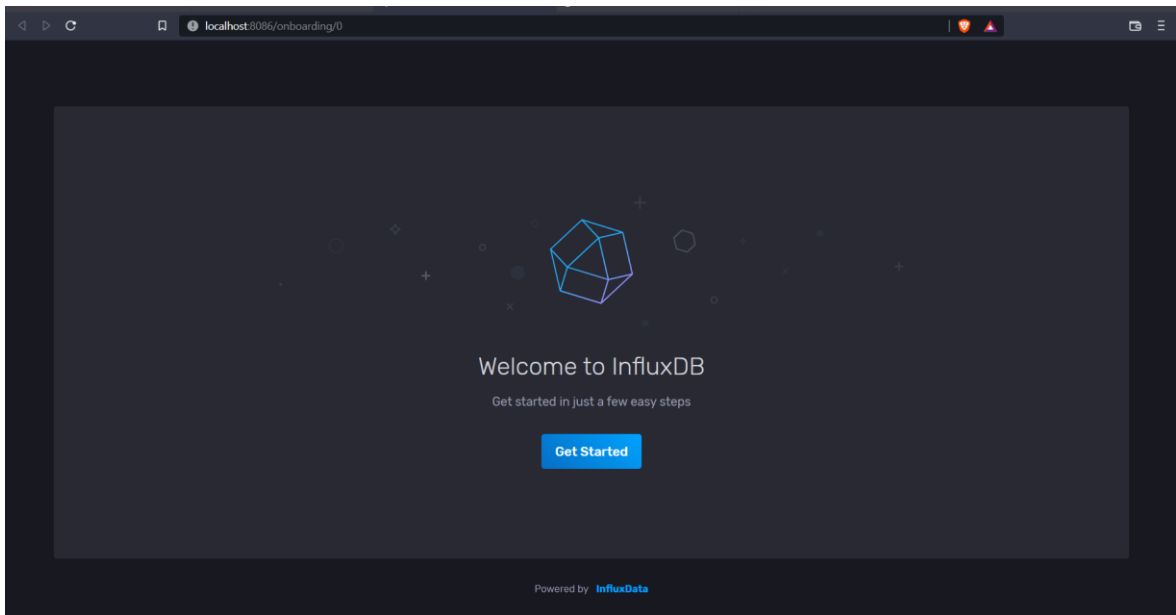
Ako je sve postavljeno i odrađeno kako treba, pokazat će se poruka

```
> influxdb.service - InfluxDB is an open-source, distributed, time series
  database
  Loaded: loaded (/lib/systemd/system/influxdb.service; enabled; vendor
  preset: enable>
  Active: active (running)
```

Za pokretanje, upisuje se komanda u terminal:

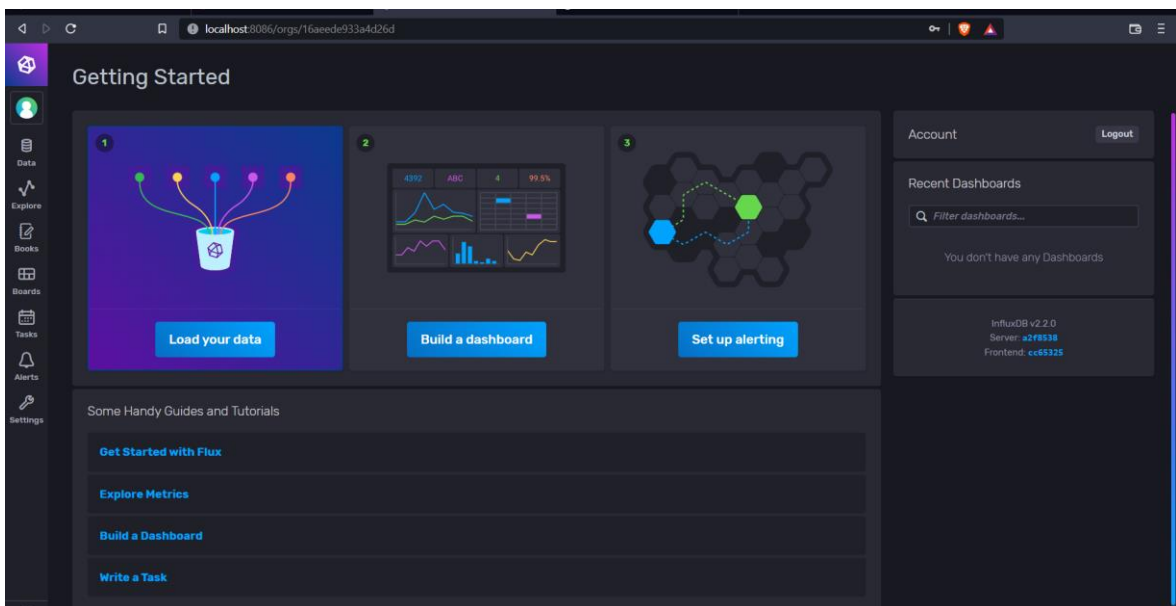
```
influxd
```

Baza podataka započinje s radom. Pristupa joj se putem lokalne web stranice na portu 8086 (<http://localhost:8086>).



Slika 22 – Početna strana InfluxDB

Nakon kratkog postavljanja i registracije, dolazimo na početnu stranu (Slika 22).



Slika 23 – Glavna strana InfluxDB

Upisivanje podataka u bazu putem Python-a odrađuje se jednostavnom instalacijom te pokretanje naredbe:

```
pip install influxdb-client
```

Inicijalizacija biblioteke u pythonu izvodi se na sljedeći način:

```
from influxdb_client import InfluxDBClient, Point, WritePrecision
from influxdb_client.client.write_api import SYNCHRONOUS

# You can generate an API token from the "API Tokens Tab" in the UI token =
"<YOUR TOKEN HERE>" org = "<ORG NAME>" bucket = "<BUCKET NAME>" with
InfluxDBClient(url="http://localhost:8086", token=token, org=org) as
client:
```

Računalni kod 9 - Inicijalizacija InfluxDB unutar postojećeg Python koda

Potrebno je prilagoditi varijable *token*, *org* i *bucket* koje se mogu pronaći na glavnoj stranici InfluxDB (Slika 23). Ako se koristi uvijek ista baza, te varijable ostaju iste te se mogu deklarirati u programu kao zasebne varijable.

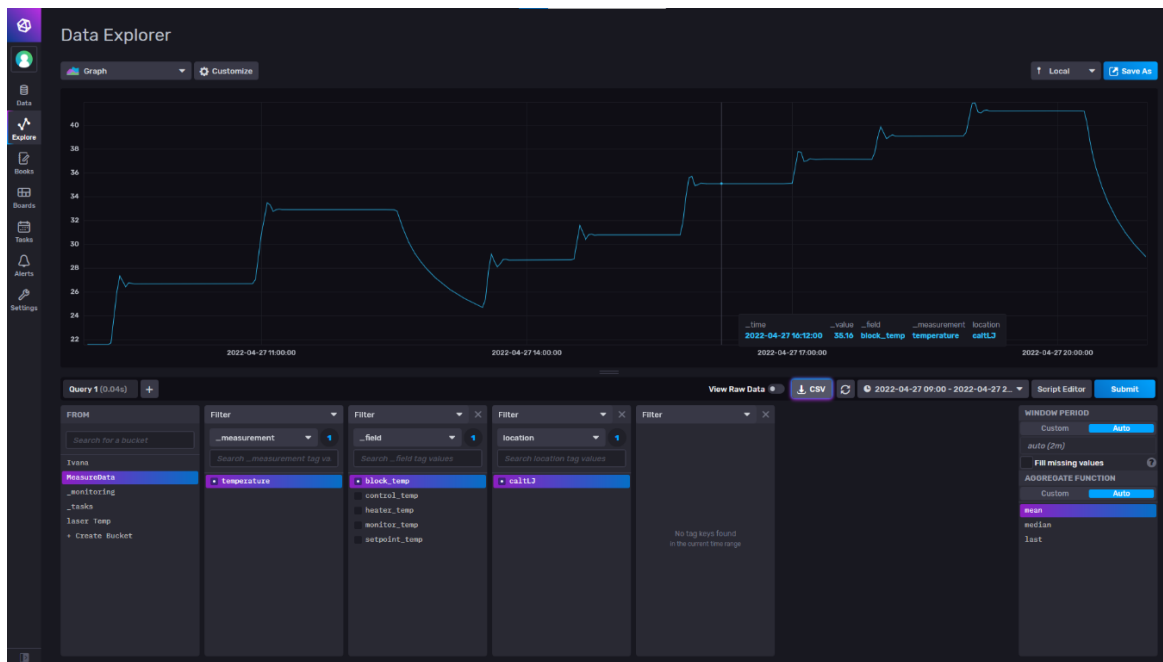
Upis u bazu također se odvija na jednostavan način. Definiramo funkciju *dbWrite* koju koristimo u svim programima.

```
def dbWrite (bucket,org,name):
    client = influxdb_client.InfluxDBClient(url=url,token=token,
                                           org=org)
    write_api = client.write_api(write_options=SYNCHRONOUS)
    p = influxdb_client.Point("temperature").tag("location",
                                                "calt").field(name,elem)
    write_api.write(bucket=bucket, org=org, record=p)
```

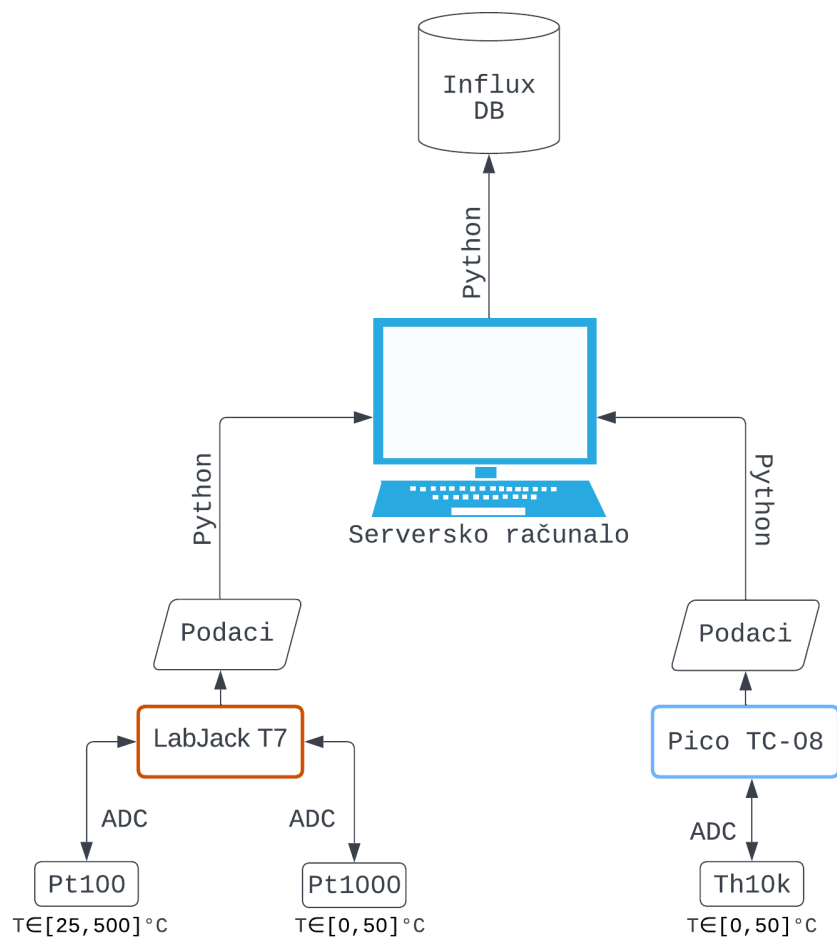
Računalni kod 10 – Funkcija za upis podataka u bazu u programskom jeziku Python

U cijeloj funkciji, potrebno je izdvojiti varijablu *p* (point) koja prima tagove i polja koja se mogu narediti po vlastitom nahođenju. U *field* objektu možemo primjetiti *elem* varijablu i kroz korišteni kod u Prilogu 1. može se uočiti da je to zapravo mjerenje koje dolazi iz uređaja.

Mjerenja koja se prikazuju u influxDB nalaze se na slici 24. Graf je interaktivan, te je za svaku točku moguće dobiti njen točan vremenski žig i vrijednost. Podatke je moguće grupirati po srednjoj vrijednosti, medijanu, te posljednjem mjerenju.



Slika 24 – Mjerenja izvedena uz pomoć postavljenih uređaja



Slika 25 - Procesna shema za postav uređaja

8 Metodički dio

NASTAVNA PRIPREMA	
Škola: Opća gimnazija Ime i prezime nastavnika: Tin Pakasin	Predmet: Informatika
	Razred: 4.
Naziv metodičke jedinice: Baze podataka	Broj sata: 1
	Nadnevak: 27.05.2022
Nastavna jedinica: Uvod u baze podataka	
Predmetni ishodi: <ul style="list-style-type: none">• A.2.4 Opisuje osnovne pojmove baze podataka	
Razrada međupredmetnih ishoda: <ul style="list-style-type: none">• Učenik samostalno ili uz navođenje može konstruirati bazu podataka• Primjenjuje jednostavne upite nad podacima• Argumentira prednosti korištenja baze podataka• Uočava njihove elemente, dijelove i relacije	
Očekivanja međupredmetnih tema: <ul style="list-style-type: none">• eDruštvo D.1.1 - Pažljivo i odgovorno barata informacijskom tehnologijom te štiti svoje podatke• Informacijske i komunikacijske tehnologije – A.4.1-Učenik kritički odabire odgovarajuću digitalnu tehnologiju	

- Informacijske i komunikacijske tehnologije C.4.1 – Učenik samostalno provodi složeno pretraživanje informacija u digitalnom okruženju
- Informacijske i komunikacijske tehnologije C.4.2 – Učenik samostalno provodi kritički procjenjuje proces, izvore i rezultate pretraživanja
- Učiti kako učiti A.5.1 – Učenik samostalno traži nove informacije iz različitih izvora, transformira ih u novo znanje i uspješno primjenjuje pri rješavanju problema
- Učiti kako učiti B.4 – Učenik prati učinkovitost učenja i napredovanje tijekom učenja

Tip sata:

Vježbe i samostalni projekt za one koji žele znati više

Oblici rada: Frontalni rad , rad u grupama ili individualni rad

Nastavne metode:

Metoda razgovora, usmjerena rasprava, kooperativno rješavanje zadataka u skupinama, praktičan rad, učenički projekti/seminari

Nastavna sredstva i pomagala:

Računalo, projektor

Za one koje žele znati više: senzor, uređaj za prikupljanje podataka

Sadržaj rada

UVODNI DIO

Motivacija:

U ovom slučaju zamislimo želimo vidjeti kako se mijenja temperatura učionice tokom dana i noći u periodu od 48 sati. Povećava li se temperatura učionice kada su u njoj učenici?

Sa učenicima pokrećem raspravu o alatima potrebnim za izvedbu takvih mjerenja.

Očekivani učenički odgovori: senzor za mjerenje temperature, računalo, i mjesto za spremanje mjerenih podataka.

Ukoliko učenici ne navedu neke od stvari, pokrećemo usmjerenu raspravu u smjeru potrebnih, a nenavedenih stvari.

Otkrivam temu današnjeg sata koja je ujedno i naslov lekcije: Uvod u baze podataka

GLAVNI DIO

Učenike pitam, što misle da je baza podataka? Čemu ona služi? Koristi li se često? Gdje se one nalaze? Prikupljam učeničke odgovore, te usmjerenom raspravom mogu ispraviti netočne pretpostavke učenika.

Mogući odgovori učenika: Baza podataka je skup podataka, služi sa spremanje podataka i koristi se često. One se nalaze u računalu. Ovdje možemo lako otkloniti učeniku miskoncepciju prije nego prijedemo na detaljniju razradu baza podataka.

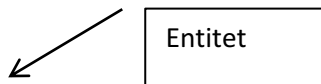
Učenicima najbliži primjer je e-Dnevnik. Možemo komentirati koji tipovi podataka čine e dnevnik.

Sa učenicima započinjem raspravu o prednostima baze podataka.

Prednosti baze podataka:

- Jednostavno skladištenje
- Brzo pretraživanje
- Jednostavan unos i brisanje
- Jednostavno upravljanje

Idejno konstruirajmo bazu podataka. Učenici daju prijedloge koje podatke ćemo prikupljati kako bi kasnije mogli obraditi podatke.



Mjerenje	
Temperatura	Atribut
Lokacija mjerenja	Atribut
Sat mjerenja	Atribut
Datum mjerenja	Atribut

Svako mjerenje odnosno entitet, prikazuje se kao tablica. A atributi se mogu opisati kao stupci tablice. Sa učenicima raspravljam zašto je baš ova podjela, možemo li dodati ili izbaciti neke od atributa i zašto?

Učenike raspodijelim u grupe. Njihov zadatak je napisati kako bi izgledala baza podataka za sustav eDnevnik. Potrebno je razvrstati elemente baze u entitete i attribute. Učenicima stojim na raspolaganju za sva njihova pitanja te komentiramo i navodimo učenike na odgovor.

Nakon završetka zadatka, predstavnik grupe učenika prezentira svoj rad na ploči te objašnjava zašto su odabrali takav način podjele baze. Ponovimo to za svaku grupu učenika te komentiramo prednosti i mane za svaki predstavljeni način. Očekujem da će učenici doći do poteškoće u rješavanju problema upisa ocjena u bazu podataka. Sa njima prolazimo način spajanja više tablica u istoj bazi podataka.

Očekivani odgovor prije poteškoća:

Učenik	Entitet	
Ime		Atribut
Prezime		Atribut
Predmet 1		Atribut
Predmet 2		Atribut
Predmet 3		Atribut
Predmet 4		Atribut

Uvodim kako izgleda rubrika dnevnika za ocjene za predmet informatika:

Predmet (informatika)		
Pismena provjera znanja	5,5,5,5	
Usmena provjera znanja	5,4,3,2	
Programiranje	5,5,5,5	
Zalaganje	1,5,5,1	

Međutim, učenici će primijetiti kako nismo riješili problem upisa u bazu podataka. Želimo moći upisivati ocjene jednu po jednu, znači da moramo bazu raščlaniti na više dijelova.

Od učenika primam sugestije kako pristupiti ovom problemu i kako bi ga oni riješili. Očekivani odgovor glasi da svaka cjelina (pisana provjera znanja, usmena provjera znanja, programiranje, zalaganje mora biti tablica sama za sebe pa crtamo novu tablicu.

Pisana provjera znanja	Entitet
Ocjena	Atribut

Usmena provjera znanja	Entitet
Ocjena	Atribut

Programiranje	Entitet
Ocjena	Atribut

Zalaganje	Entitet
Ocjena	Atribut

Kada smo raščlanili tablice, kako ćemo znati kome te ocjene pripadaju? Preostaje nam identificirati učenika. Pošto su učenici u dnevniku poredani po abecedi, svakom učeniku je dodijeljen redni broj. To je stvar koja nam nedostaje. U staru tablicu upisujemo novi redak.

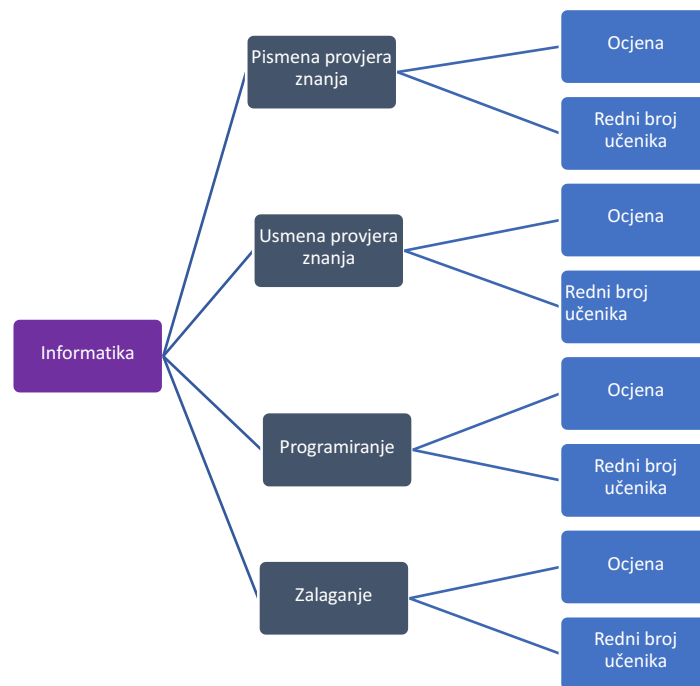
Pisana provjera znanja	Entitet
Redni broj učenika	Atribut
Ocjena	Atribut

Usmena provjera znanja	Entitet
Redni broj učenika	Atribut
Ocjena	Atribut

Programiranje	Entitet
Redni broj učenika	Atribut
Ocjena	Atribut

Zalaganje	Entitet
Redni broj učenika	Atribut
Ocjena	Atribut

Završni izgled baze u obliku grafa crta se na ploču.



ZAVRŠNI DIO

Sa učenicima prolazim ponavljanje današnjeg sata, u frontalnom načinu rada. Postavljaju se pitanja:

- Što su baze podataka? Baze podataka je skup podataka.
- Čemu služe baze podataka? Za pohranu velikih količina podataka.
- Koje su prednosti baze podataka? Prednosti baze podataka: jednostavno skladištenje, brzo pretraživanje, jednostavan unos i brisanje, Jednostavno upravljanje
- Koja su dva glavna imena dijelova baze podataka? Atributi i entiteti

Za kraj, za one koji žele znati više, moguće je implementirati motivacijsko pitanje sa uvodnog dijela sata skupa sa profesorom: Implementacija mjerenja temperature uz pomoć baze podataka. Za ostatak razreda, dodjeljuje se domaća zadaća da se raspiše isti proces izrade baze podataka za neki drugi nastavni predmet (npr. fizika). Potrebno je komentirati mogućnosti poboljšanja baze podataka obrađene u ovom školskom satu.

9 Zaključak

U radu je proveden eksperimentalni dio automatizacije prikupljanja podataka uz pomoć Python programskog jezika i InfluxDB baze podataka. Uređaji i senzori koji su bili korišteni su: Labjack T7, Pico TC-08, Pt100 i Pt1000 RTD senzori i Th10k termistor.

Zadatak je uspješno obavljen, te se podaci nastavljaju prikupljati i nakon završetka rada. Znanstvenici u laboratoriju nastaviti će koristiti do sad napisane programe te će ih nadograđivati po potrebi. Plan za budućnost bit će dodavanje dodatnih mjerenja na različitim mjestima u laboratoriju, te proširivanje mjerenja na tlak u prostoriji.

Preporuka je da se izbjegava korištenje Pico uređaja radi manja dokumentacije, nespretne implementacije za korištenje i slabe dokumentacije koja je dostupna na web stranicama samog proizvođača.

LabJack T7 s druge strane je intuitivan i jednostavan za postavljanje, dolazi sa više opcija i mogućnosti, te ima širu lepezu mogućih mjerenja. Python se pokazao kao odličan izbor programskog jezika radi svoje jednostavnosti i mogućnosti brzog pisanja samog koda za obradu podataka.

InfluxDB se iskazao kao jednostavan, ali moćan alat čija automatska obrada agregacije uvelike olakšava prikupljanje podataka. Dodatak grafičkog sučelja koje u stvarnom vremenu prikazuje podatke, crta graf i može ukazivati na odstupanja od očekivanih mjerenja uvelike olakšava rad s bazom podataka. Jezik postavljanja upita je sličan SQL, a dokumentacija je čitka i opširna sa stvarnim primjerima koji najčešće daju odgovor na specifično pitanje koje se traži.

Umjeravanje senzora dalo je očekivane rezultate. Svi senzori bez detaljnog proučavanja imaju nevelik odmak od stvarne vrijednosti mjerenja, ali čak i sa takvim problemom vidimo da su i dalje precizni i točni u odnosu na mjerenja za koja se koriste. Detaljnijom analizom rada i uređaja i senzora mogli bi se otkloniti odmaci te dobiti ekstremno precizni izračuni koji bi u budućnosti mogli biti potrebni laboratoriju. Potrebno bi bilo napraviti bolju kontrolu varijabli ako će se pristupati poboljšanju točnosti senzora.

10 Literatura

- [1] Python Software Foundation, »Python Documentation,« [Mrežno]. Available: <https://docs.python.org/3/faq/general.html#what-is-python>. [Pokušaj pristupa 5 Svibanj 2022].
- [2] R. Manger, Baze Podataka, Zagreb, 2021.
- [3] E. Codd, »A relational Model of Data for Large Shared Data Banks,« u *Volume 13/Number 6*, 1970.
- [4] R. Manger, »Srce,« 2010.. [Mrežno]. Available: https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/d310_polaznik.pdf.
- [5] E. Musa, G. Delač, M. Šilić i K. Vladimir, »Comparison of Relational and Time-Series Databases for Real-Time Massive Datasets,« u *MIPRO*, 2019..
- [6] Labjack, »Labjack, Inc.,« 2022. [Mrežno]. Available: <https://labjack.com/support/datasheets/t-series>. [Pokušaj pristupa 12 5 2022].
- [7] Pico Technology, »Thermocouple Data Logger Datasheet,« [Mrežno]. Available: <https://www.picotech.com/download/datasheets/usb-tc-08-thermocouple-data-logger-data-sheet.pdf>. [Pokušaj pristupa 22 05 2022].
- [8] Pico Technology, »TC-08 Data Logger,« [Mrežno]. Available: <https://www.picotech.com/data-logger/tc-08/thermocouple-data-logger>. [Pokušaj pristupa 22 05 2022].
- [9] A. Smontara, »Termočlanci kao senzori temperature,« *Matematičko fizički list*, pp. 97-99, 2002.
- [10] R. Malarić, »Fakultet elektrotehnike i računalstva, Mjerenja u tehnološkim procesima,« 2015.. [Mrežno]. Available: https://www.fer.unizg.hr/_download/repository/skripta_k_hr_N.pdf.

- [11] LabJack - Measurement and Automation, »LabJack Products - T7,« [Mrežno]. Available: <https://labjack.com/products/t7>. [Pokušaj pristupa 22 05 2022].
- [12] G. Strouse, »On the Increase of Electrical resistance in Conductors with Rise of Temperature, and Its Application to the Measure of Ordinary and Furnace Temperature; Also on Simple Method of Measuring Electrical Resistance,« 2009.
- [13] Engineers Community, »Construction of RTD,« [Mrežno]. Available: <https://engineerscommunity.com/t/construction-of-rtd/6686>. [Pokušaj pristupa 22 05 2022].
- [14] RealPars, »RealPars Corporation,« 2022. [Mrežno]. Available: <https://realpars.com/pt100/>. [Pokušaj pristupa 22 05 2022].
- [15] Hrvatska Enciklopedija, »Termistor,« [Mrežno]. Available: <https://www.enciklopedija.hr/Natuknica.aspx?ID=60958>. [Pokušaj pristupa 22 05 2022].
- [16] »Struktura elektronskih vrpca,« [Mrežno]. Available: https://hr.wikipedia.org/wiki/Metalna_veza#/media/Datoteka:Struktura_elektronskih_vrpca.jpg. [Pokušaj pristupa 22 05 2022].
- [17] Microsoft, »Microsoft,« 2021. [Mrežno]. Available: <https://docs.microsoft.com/en-us/cpp/atl/reference/chandle-class?view=msvc-170>.
- [18] LabJack - Measurement and Automation, »T7 Quickstart Tutorial,« [Mrežno]. Available: <https://labjack.com/support/quickstart/t7>. [Pokušaj pristupa 22 05 2022].
- [19] LabJack - Measurement and Automation, »Python for LJM - Windows, Mac, Linux,« [Mrežno]. Available: <https://labjack.com/support/software/examples/ljm/python>. [Pokušaj pristupa 22 05 2022].
- [20] LabJack - Measurement and Automation, »LJM Software Installers - T4, T7, Digit,« [Mrežno]. Available: <https://labjack.com/support/software/installers/ljm>. [Pokušaj pristupa 22 05 2022].

- [21] Encyclopedia Britannica, »Latent Heat,« [Mrežno]. Available: <https://www.britannica.com/science/latent-heat>. [Pokušaj pristupa 22 05 2022].
- [22] openstax, »Phase Change and Latent Heat,« [Mrežno]. Available: <https://openstax.org/books/college-physics/pages/14-3-phase-change-and-latent-heat>. [Pokušaj pristupa 22 05 2022].
- [23] Cmglee, »Wikipedia,« [Mrežno]. Available: <https://commons.wikimedia.org/w/index.php?curid=14939155>.
- [24] Pentronic, Pnetronic, 2016. [Mrežno]. Available: <https://www.pentronic.se/wp-content/uploads/2019/05/15-2-what-affects-a-pt100-sensors-precision.pdf>. [Pokušaj pristupa 17 5 2022].
- [25] Guilcor, »Temperature sensors,« [Mrežno]. Available: <https://www.guilcor.com/content/30-classe-de-precision-des-sondes-pt1000>. [Pokušaj pristupa 22 05 2022].
- [26] ThorLabs, »TH10K Thermistor Description,« 2018. [Mrežno]. Available: <https://www.thorlabs.com/drawings/819f3f36a1e301fc-3CAEDC56-DF5E-21E3-1D161418F356F2DB/TH10K-SpecSheet.pdf>. [Pokušaj pristupa 22 05 2022].
- [27] influxdata, »InfluxData Documentation - Install InfluxDB,« [Mrežno]. Available: <https://docs.influxdata.com/influxdb/v2.2/install/?t=Linux>. [Pokušaj pristupa 22 05 2022].
- [28] C. Date, An Introduction to Database Systems, Addison-Wesley, 2000.
- [29] B. Y. Ludow, »Optical atomic Clocks,« 10 February 2015.