

Vizualizacija podataka pomoću InfluxDB na primjeru podataka iz Tesla Wall Connectora

Ostović, Martina

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, School of Applied Mathematics and Informatics / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet primijenjene matematike i informatike**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:126:644032>

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

Download date / Datum preuzimanja: **2025-03-10**



mathos

Repository / Repozitorij:

[Repository of School of Applied Mathematics and Informatics](#)



DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJ



SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET PRIMIJENJENE MATEMATIKE I INFORMATIKE

Sveučilišni prijediplomski studij Matematika i računarstvo

Vizualizacija podataka pomoću InfluxDB na primjeru podataka iz Tesla Wall Connectora

ZAVRŠNI RAD

Mentor:

**izv. prof. dr. sc. Danijel Graho-
vac**

Student:

Martina Ostović

Osijek, 2024.

Sadržaj

1	Uvod	1
2	Tesla Wall Connector data logger	3
3	InfluxDB	5
3.1	InfluxDB Cloud	5
3.1.1	Organizacija podataka	6
3.1.2	Administracija	6
3.1.3	Python klijentska biblioteka	9
3.1.4	Vizualizacija podataka	10
4	Vizualizacija podataka iz punjača	13
5	Implementacija	19
5.1	Organizacija koda	19
5.2	GitHub repozitorij	19
	Literatura	21
	Sažetak	23
	Summary	25
	Životopis	27

1 | Uvod

U današnje vrijeme raste potreba za održivim razvojem zbog boljeg utjecaja na okoliš. Iz tog razloga sve je veća potražnja za električnim automobilima u odnosu na automobile sa unutarnjim izgaranjem. Vozačima električnih automobila je važno da koriste punjač koji omogućuje brzo, jeftino i praktično punjenje. Kućni punjači poput Tesla Wall Connector Gen 3 za električne automobile su idealno rješenje za to.

Kako bi vlasnici električnih automobila imali mogućnost uvida u status punjača izrađen je projekt Tesla Wall Connector data logger koji omogućuje analizu prikupljenih podataka poput potrošnje električne energije tijekom sesije, cijene potrošene električne energije tijekom sesije, trajanja punjenja, jakosti struje itd., čiji će dio s vizualizacijom tih podataka biti pobliže opisan u radu. Cilj ovog projekta bilo je razviti sustav koji se relativno jednostavno instalira na mikro računalo Raspberry Pi u lokalnoj WiFi mreži, neprekidno dohvaća podatke iz punjača te omogućuje grafički pristup povijesnim i trenutnim podacima u InfluxDB Cloudu.

U drugom poglavlju će kratko biti opisan projekt i tehnologije korištene u radu. Treće poglavlje će detaljno opisati InfluxDB platformu, dok će četvrto poglavlje detaljno opisati vizualizaciju podataka iz Tesla Wall Connector Gen 3 punjača pomoću InfluxDB Clouda. Peto poglavlje će opisati organizaciju koda projekta.

2 | Tesla Wall Connector data logger

Kako Tesla Wall Connector Gen 3 punjač može biti spojen na lokalnu WiFi mrežu, kroz nedokumentirani REST API se omogućava prikupljanje podataka o statusu punjača. Nakon instalacije data loggera na Raspberry Pi mikro računalo automatski se pokreće Python skripta koja omogućava prikupljanje podataka s punjača i slanje tih podataka u InfluxDB Cloud. InfluxDB Cloud platforma omogućuje vizualizaciju dohvaćenih podataka i praćenje statistike podataka. Cilj ovog projekta bilo je razviti sustav koji se relativno jednostavno instalira na mikro računalo RPi u lokalnoj WiFi mreži, neprekidno dohvaća podatke iz punjača te omogućuje grafički pristup povijesnim i trenutnim podacima u InfluxDB Cloudu.

Python skripta projekta, koja prikuplja i pohranjuje podatke iz punjača, je na mikro računalu Raspberry Pi 3 Model B. Projekt je izrađen koristeći programski jezik Python i InfluxDB Cloud bazu podataka vremenskih nizova.

3 | InfluxDB

InfluxDB je platforma koju je razvila softverska tvrtka InfluxData. To je najpopularnija baza podataka otvorenog koda za korisnike koji upravljaju podacima vremenskih nizova. Visokih je performansi [14]. Omogućuje efikasnu analitiku podataka u stvarnom vremenu i prikupljanje velike količine podataka s preciznošću od nanosekunde. Idealna je za praćenje statistike podataka. Efikasno obrađuje upit podataka vremenskih nizova te brzo pohranjuje veliku količinu podataka [15], [2]. Omogućava skaliranje resursa prema potrebama korisnika. Ima mogućnost automatskog brisanja podataka ako je njihovo trajanje starije od navedenog. To pomaže upravljanju pohranom te osigurava da stari i nepotrebni podaci ne troše resurse [14]. Da bi analitika podataka bila jednostavnija, InfluxDB omogućuje pristup raznim alatima za vizualizaciju podataka poput InfluxDB UIa (eng. *user interface*), Chronografa, Grafane i ostalih [12], [1], [5]. Kako bi se zaštitili podaci, InfluxDB pruža sigurnosne mehanizme. Podržava više programskih jezika i upitnih jezika tako da nudi klijentske biblioteke za mogućnost povezivanja željene aplikacije ili skripte s InfluxDB i razne druge integracije [14]. Idealna je za praćenje financijskih podataka, senzorskih podataka, performansi IT mreža, podataka u vezi okoliša, medicinskih podataka i mnogo drugih podataka vremenskih nizova [2], [13].

Razvijene su samoupravljajuće InfluxDB verzije:

- InfluxDB Clustered
- InfluxDB Enterprise
- InfluxDB OSS

te InfluxDB verzije kojim upravlja tvrtka InfluxData:

- InfluxDB Cloud Serverless
- InfluxDB Cloud
- InfluxDB Cloud Dedicated [6].

3.1 InfluxDB Cloud

Za potrebe projekta Tesla Wall Connector data logger je korišten InfluxDB Cloud pa će zato u ovome poglavlju biti detaljno opisan.

3.1.1 Organizacija podataka

InfluxDB Cloud model podataka organizira podatke vremenskih nizova u „buckets” i „measurements” na sljedeći način:

- „**bucket**” je imenovana lokacija na kojoj su pohranjeni podaci vremenskih nizova. Kombinira koncept baze podataka i razdoblje zadržavanja podataka. Može sadržavati više „measurements”.
- „**measurement**” je logičko grupiranje za podatke vremenskih nizova. Sadrži više „tagova” i „fieldova”.
- „**point**” je pojedinačni zapis podataka identificiran svojim „measurementom”, „fieldovima”, vremenskom oznakom i opcionalno „tagovima”.

„Bucket” pripada radnom prostoru za grupu korisnika, tj. organizaciji. Svi podaci spremljeni u InfluxDB u `_time` stupcu imaju **vremensku oznaku**. InfluxDB prikazuje vremensku oznaku u formatu na primjer: 2022-01-01T00:00:00.00Z. „**Fieldovi**” su ključevi podataka zajedno sa njihovim odgovarajućim vrijednostima, gdje se vrijednosti mijenjaju s vremenom. Nisu indeksirani. „**Tagovi**” su ključ-vrijednost parovi namijenjeni za pohranu metapodataka za svaki „point” te su indeksirani. Pomažu u brzom filtriranju podataka. Koriste se na primjer za identifikaciju izvora podataka poput lokacije, hosta i drugo. Podaci u `_measurement` stupcu imaju ime „measurements”, u stupcu `_field` ključ „fields” te njegovu vrijednost u stupcu `_value`. „Point” predstavlja neki događaj u bazi podataka vremenskih nizova [4], [7]. Na Slici 3.1 je prikazan podatak „`contactor_closed_str`” u tablici u InfluxDB Cloudu gdje se u svakom redu vidi jedan „point”.

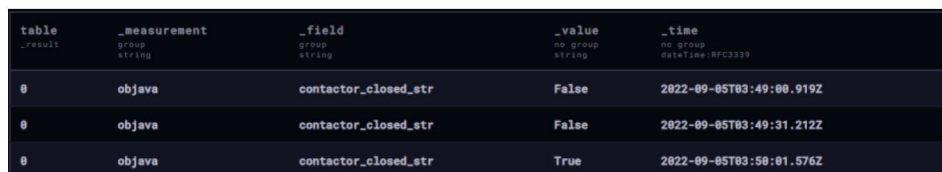


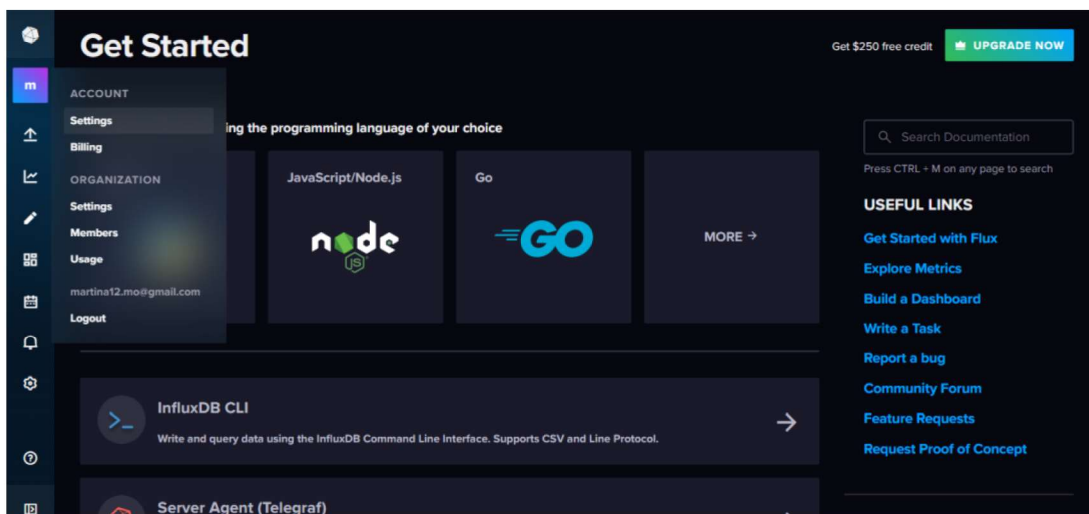
table	_measurement	_field	_value	_time
..result	group string	group string	no group string	no group dateTime:RFC3339
0	objava	contactor_closed_str	False	2022-09-05T03:49:00.919Z
0	objava	contactor_closed_str	False	2022-09-05T03:49:31.212Z
0	objava	contactor_closed_str	True	2022-09-05T03:50:01.576Z

Slika 3.1: Podatak prikazan u tablici u InfluxDB Cloudu.

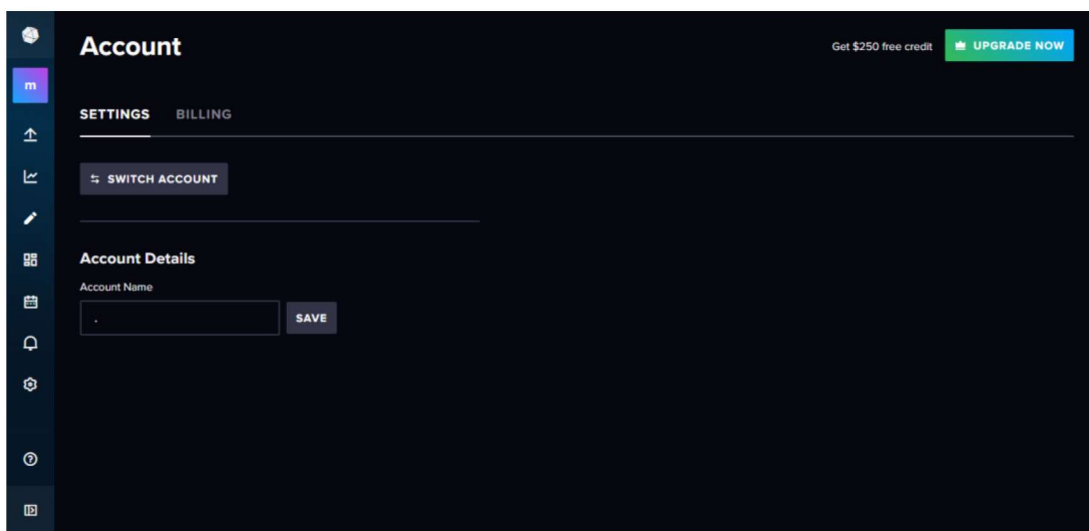
3.1.2 Administracija

Kao što je napisano u prethodnom potpoglavlju organizacija je radni prostor za grupu korisnika. Svi dashboardi, „bucketi”, članovi itd., pripadaju organizaciji. Korisnički račun može sadržavati više organizacija. Organizacije se mogu koristiti za odvajanje podataka, timova, davatelja usluga, regija i okruženja (*dev*, *staging*, *prod*) unutar istog korisničkog računa. Za pregled organizacija se koristi InfluxDB korisničko sučelje (UI) 3.2 ili influx CLI (eng. *command line interface*). Koristeći influx CLI se to može napraviti naredbom `influx org list`. Svaka organizacija ima svoj ID. Za ažuriranje naziva organizacije se također koristi InfluxDB UI ili influx CLI. Moguće je obrisati organizaciju. Ako se koristi besplatni plan to se može obaviti u InfluxDB UIu tako da se klikne ikona korisnika u lijevom izborniku, a zatim

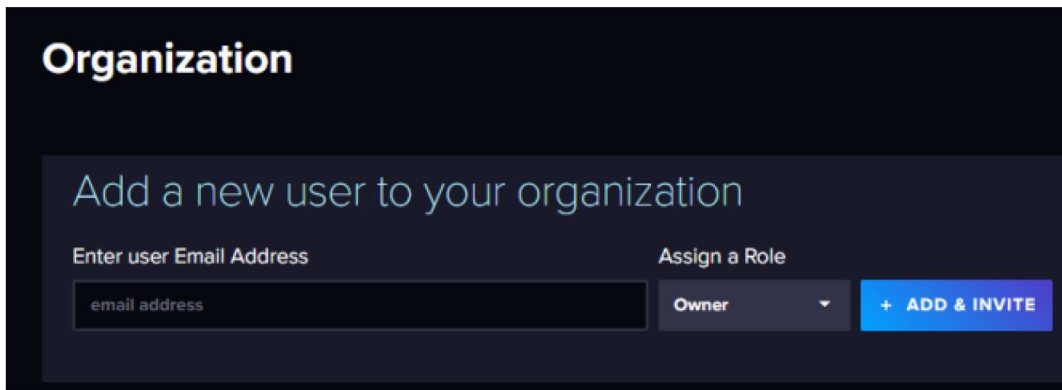
„Settings“ u padajućem izborniku i odabere „Delete Organization“ te klikne „DELETE“. InfluxDB Cloud omogućava da se pozovu korisnici u organizaciju i surađuje s njima. Upravljanje korisnicima organizacije se obavlja u InfluxDB UIu na način da se u padajućem izborniku sa Slike 3.2 odabere „Members“. Pozvati korisnika u organizaciju je moguće pod „Add a new user to your organization“ tako da se upiše njegova email adresa i odabere njegova uloga u organizaciji, a zatim klikne „+ ADD & INVITE“ (Slika 3.4). Tada je pozivnica sa aktivacijskim linkom poslana na određenu email adresu. Taj link traje 72 sata. Kada je link aktiviran korisnik je dodan u organizaciju. Korisnici moraju biti pozvani jedan po jedan. Moguće je opet poslati pozivnicu, povući ju i maknuti korisnika iz organizacije. Omogućeno je kopiranje organizacije [10].



Slika 3.2: InfluxDB UI.

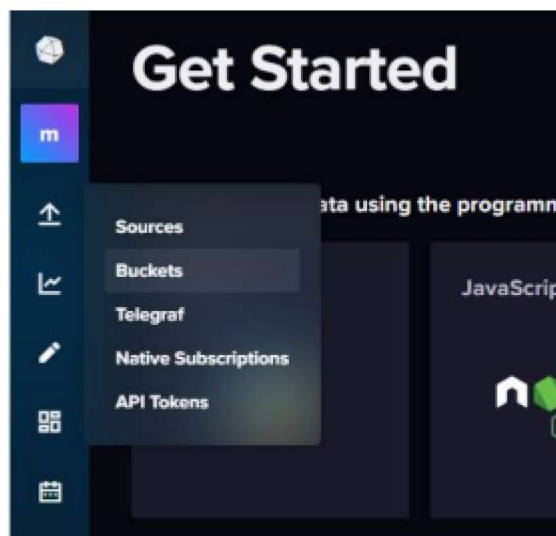


Slika 3.3: Korisnički račun.



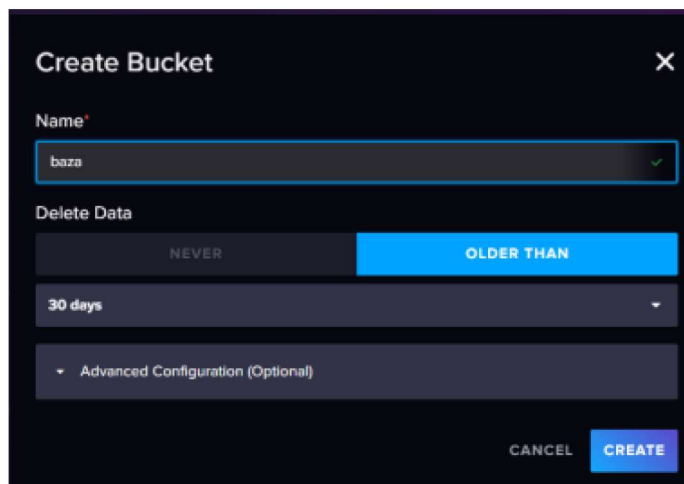
Slika 3.4: Pozivanje korisnika u organizaciju.

Kako bi se podaci vremenskih nizova pohranili u InfluxDB Cloud moguće je napraviti „bucket“. Kreirati i ažurirati „bucket“ te pregledati listu „bucketa“ s pripadajućim nazivom i periodom zadržavanja podataka te ID-em je moguće koristeći InfluxDB UI, influx CLI ili InfluxDB HTTP API. To se u InfluxDB UIu može tako da se odabere „Load Data“ s lijevog izbornika i zatim „Buckets“ (Slika 3.5). Za kreiranje „bucketa“ odabere se „+ Create Bucket“, a zatim se upiše ime „bucketa“ i odabere pripadajući period zadržavanja podataka (Slika 3.6). Moguće je kreirati „bucket“ odabirom „Data Explorer“ s lijevog izbornika ili tijekom uređivanja ćelije dashboard-a na isti način. InfluxDB briše „pointove“ sa vremenskim oznakama starijim od perioda zadržavanja podataka. Ažuriranje naziva „bucketa“ u InfluxDB UIu je moguće tako da se pritisne „SETTINGS“ te nakon toga „RENAME“. Za ažuriranje pripadajućeg perioda zadržavanja podataka umjesto „RENAME“ se pod „Delete Data“ odabere željeni period. Brisanje „bucketa“ je također moguće koristeći InfluxDB UI ili influx CLI [9].



Slika 3.5: Kreiranje „bucketa“.

InfluxDB API token osigurava sigurnu interakciju između korisnika i podataka. Token pripada organizaciji i identificira InfluxDB odobrenja unutar organizacije.



Slika 3.6: Kreiranje „bucketa” i određivanje perioda zadržavanja „pointova”.

Token za puni pristup omogućuje puni pristup za čitanje i pisanje svim resursima u organizaciji. Čitaj/piši token omogućuje pristup za čitanje ili pisanje ili oboje specifičnim „bucketima” u organizaciji. Kreiranje, pregled, ažuriranje i brisanje tokena je moguće koristeći InfluxDB UI, influx CLI ili InfluxDB API. Kako bi se moglo upravljati tokenima u InfluxDB UI-u potrebno je odabrati „Load Data” s lijevog izbornika te nakon toga „API Tokens”. Za kreiranje tokena za puni pristup treba kliknuti „+ Generate API Token” te odabrati „All Access API Token”. Moguće je napraviti prilagođeni token te klonirati token. Korisnik može vidjeti token status i popis dopuštenja pristupa tako da klikne opis tokena u listi tokena. Za ažuriranje opisa tokena je potrebno kliknuti na ikonu olovke pored naziva tokena u „Description” rubrici. Također se može omogućiti ili onemogućiti token. Token se mogu koristiti za autentikaciju zahtjeva prema InfluxDBu, uključujući zahtjeve za pisanje i upit podataka te upravljanje podacima i resursima. To je moguće koristeći influx CLI, API zahtjeve napravljene sa klijentskim bibliotekama ili alate poput curl [8].

3.1.3 Python klijentska biblioteka

InfluxDB Cloud nudi Python klijentsku biblioteku za mogućnost povezivanja željene aplikacije ili skripte s InfluxDB Cloud. Prvo se mora instalirati Python klijentska biblioteka, a jedan on načina je pisanje naredbe `pip install influxdb-client`, te se mora osigurati da je InfluxDB Cloud pokrenut. Slanje podataka u InfluxDB Cloud je moguće tako da korisnik naredbom `import` u svojem Python programu učita InfluxDB Python klijentsku biblioteku (Kod 3.1).

```
1 import influxdb_client
2 from influxdb_client.client.write_api import SYNCHRONOUS
```

Kod 3.1: Učitavanje Python klijentske biblioteke.

Nakon toga je potrebno definirati u svojem Python programu varijable `bucket`,

org, token i url sa pripadnim nazivima. InfluxDBClient klasa služi za povezivanje

```
url = "https://europe-west1-1.gcp.cloud2.influxdata.com"
```

Slika 3.7: Primjer url-a.

sa InfluxDB Cloud. InfluxDBClient objekt ima parametre url, token i org što su podaci za autentikaciju (Kod 3.2).

```
1 client = influxdb_client.InfluxDBClient(  
2     url=url,  
3     token=token,  
4     org=org  
5 )
```

Kod 3.2: Instanciranje klienta.

Objekt InfluxDBClient ima *write_api* metodu za konfiguraciju (Kod 3.3).

```
1 write_api = client.write_api(write_options=SYNCHRONOUS)
```

Kod 3.3: Metoda *write_api*.

Potrebno je kreirati Point objekt kako bi se podaci upisali u InfluxDB Cloud (Kod 3.4).

```
1 t = influxdb_client.Point("temperatura").tag("location", "0sijek").  
    field("temperature", 32)  
2 write_api.write(bucket=bucket, org=org, record=t)
```

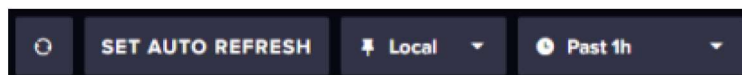
Kod 3.4: Primjer upisivanja podataka u InfluxDB Cloud.

Python klijentska biblioteka podržava pravljenje upita podataka koristeći Flux [11]. Flux je funkcionalni skriptni jezik otvorenog koda namijenjen za rad s podacima kojeg je razvila tvrtka InfluxData. Dizajniran je za objedinjavanje upita, obrade i analize podataka i djelovanja na podacima u jednu sintaksu [3].

3.1.4 Vizualizacija podataka

U ovome potpoglavlju će biti pobliže opisana vizualizacija podataka u InfluxDB UIu. Moguće je pretraživati i vizualizirati podatke odabirom „Data Explorer“ s lijevog izbornika, gdje je također moguće koristiti Flux. InfluxDB UI pruža alate za izgradnju dashboarda da bi se podaci mogli vizualizirati. Kako bi se upravljalo dashboardima potrebno je odabrati „Dashboards“ s lijevog izbornika. Moguće je napraviti novi, uvesti, izvesti, klonirati ili obrisati dashboard te ga imenovati. Kreirati novi dashboard se može tako da se odabere "Create Dashboard" te opcija "New Dashboard". U dashboard se mogu dodavati ćelije i imenovati ih. Svaka ćelija može biti sagrađena na temelju nekog od prikupljenih podataka. Kako bi to

bilo moguće pri uređivanju ćelija je potrebno odabrati željeni „bucket“, „measurement“ i „field“. Za umetanje postojećeg dashboarda treba odabrati opciju „Import Dashboard“ te umetnuti odgovarajuću JSON datoteku. Omogućen je odabir vremenskog intervala za koji želimo grafički prikazati podatke u dashboardu te brzine osvježavanja i vremenske zone za dashboard (Slika 3.8). Da bi se dashboard mogao izvesti potrebno je kliknuti ikonu zupčanika na dashboardu te odabrati „Export“. Nakon toga treba odabrati preuzimanje dashboarda kao JSON dokument ili neku drugu željenu opciju. Svaki dashboard ima svoj ID. InfluxDB UI pruža sljedeće grafičke prikaze u ćeliji: Graph, Band, Gauge, Graph + Single Stat, Heatmap, Histogram, Mosaic, Scatter, Single Stat, Table i Map [12].



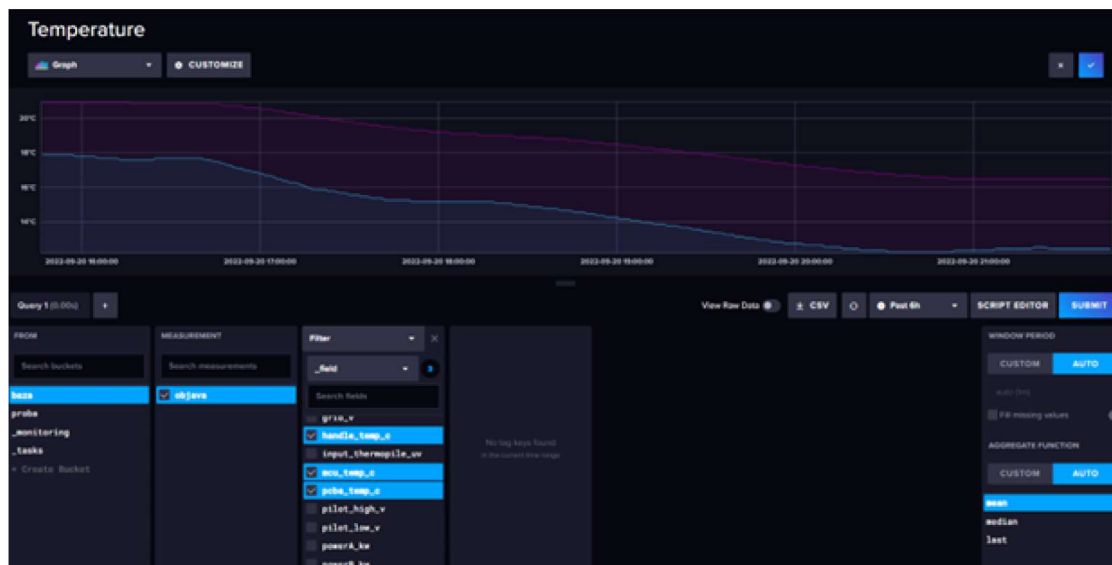
Slika 3.8: Primjer brzine osvježavanja, vremenske zone i vremenskog intervala dashboarda.

4 | Vizualizacija podataka iz punjača

Vizualizacija podataka iz Tesla Wall Connector Gen 3 punjača je ostvarena u InfluxDB korisničkom sučelju (UI) u InfluxDB Cloudu. Kreiran je „bucket“ naziva „baza“ da bi se podaci vremenskih nizova iz punjača mogli pohraniti u InfluxDB Cloud (Slika 3.6). Kako bi ti podaci bili grafički prikazani napravljen je dashboard naziva „Tesla Wall Connector Gen 3“ (Slika 4.1). To je dashboard koji se sastoji od 17 ćelija. Svaka ta ćelija je sagrađena od nekog prikupljenog podatka iz punjača da bi praćenje statistike tih podataka bilo jednostavnije. Kako bi to bilo moguće pri uređivanju ćelija je bilo potrebno odabrati „bucket“ naziva „baza“, „measurement“ naziva „objava“ i željeni „field“ (Slika 4.2).

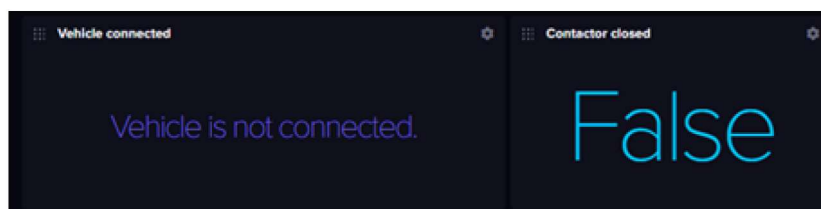


Slika 4.1: Dashboard.



Slika 4.2: Prikaz uređivanja ćelije.

Slika 4.3 prikazuje ćeliju **Vehicle connected** koja je sagrađena od podatka „vehicle_connected” čija je vrijednost iskazana kao „Vehicle is connected”/„Vehicle is not connected”. „Vehicle is connected” znači da je punjač priključen u automobil, a „Vehicle is not connected” znači da nije. Slika 4.3 također prikazuje ćeliju **Contactor closed** koja je sagrađena od podatka „contactor_closed_str” čija je vrijednost iskazana kao „True”/„False” gdje „True” znači da punjač puni automobil, dok „False” znači da punjač ne puni automobil.



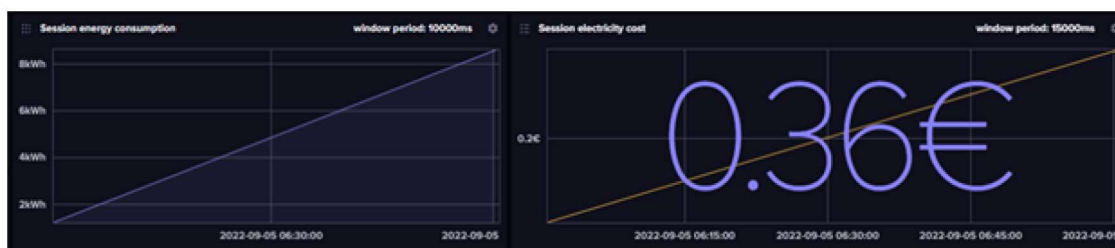
Slika 4.3: Ćelije dashboarda dok punjač ne puni automobil.

Slika 4.4 prikazuje ćeliju **Wifi connected** koja je sagrađena od podatka „wifi_connected_str” čija je vrijednost iskazana kao „True”/„False”. „True” znači da je punjač spojen na Wifi, a „False” da nije. Slika 4.4 prikazuje i ćeliju **Wifi signal strength** sagrađenu od podatka „wifi_signal_strength” čija je vrijednost iskazana u postocima. Graf u toj ćeliji prikazuje jačinu Wifi signala.

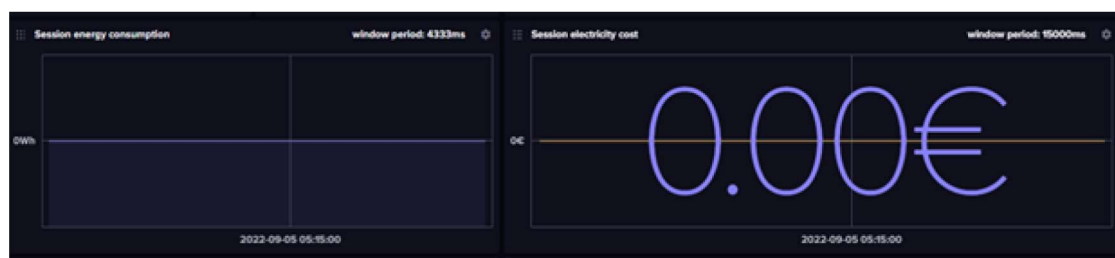


Slika 4.4: Prikaz podataka o Wifi povezanosti.

Na Slici 4.5 je prikazana ćelija **Session energy consumption** sagrađena od podatka „session_energy_wh” čija je vrijednost iskazana u Wh mjernoj jedinici. Ta ćelija prikazuje ukupnu električnu energiju koju je punjač potrošio tijekom zadnje sesije. Nakon završetka sesije punjač zadržava podatak o potrošnji električne energije tijekom zadnje sesije, te se tek pri početku nove sesije vrijednost resetira na nulu. Ista Slika također prikazuje ćeliju **Session electricity cost** sagrađene od podatka „electricity_cost” čija je vrijednost iskazana u eurima. Ta ćelija predstavlja cijenu električne energije potrošene tijekom zadnje sesije. Cijena je izračunata po jeftinijoj tarifi potrošene električne energije po satu. Nakon završetka sesije punjač zadržava podatak o cijeni zadnje sesije, tek pri početku nove sesije se cijena resetira na nulu.



Slika 4.5: Prikaz potrošnje električne energije po satu tijekom punjenja uz pripadnu cijenu.



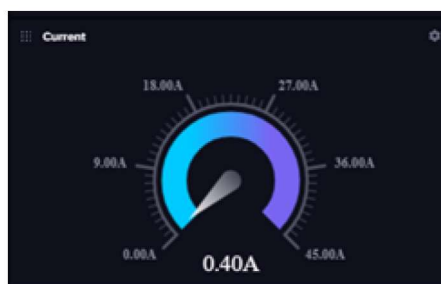
Slika 4.6: Prikaz potrošnje električne energije po satu prije punjenja uz pripadnu cijenu.

Slika 4.7 prikazuje ćeliju **Current** sagrađenu od podatka „vehicle_current_a” čija je vrijednost iskazana u amperima. Ta ćelija predstavlja jakost električne struje punjača. Slika 4.9 prikazuje ćeliju **Voltage** koja je sagrađena od podataka „voltageA_v”, „voltageB_v” i „voltageC_v” čije su vrijednosti iskazane u voltima te koja predstavlja napon punjača. Ćelija **Grid voltage** je sagrađena od podatka „grid_v” čija je vrijednost iskazana u voltima te predstavlja napon mreže (Slika 4.11).

Slika 4.2 prikazuje ćeliju **Temperature** sagrađenu od podataka „handle_temp_c” što je temperatura ručke punjača, „mcu_temp_c” što je temperatura mikrokontrolera u punjaču i „pcba_temp_c” što je temperatura tiskane pločice punjača. Vrijednost ovih podataka je iskazana u Celzijevim stupnjevima. Ćeliju **Session charging time (hours)** koja je sagrađena od podatka „session” čija je vrijednost iskazana u formatu „sati:minute:sekunde” prikazuje Slika 4.12. Ta ćelija predstavlja vrijeme koliko dugo je punjač uključen u automobil u zadnjoj sesiji (punjač ne mora to cijelo vrijeme puniti, nego je od tada priključen u automobil).



Slika 4.7: Prikaz jakosti električne struje punjača tijekom punjenja.



Slika 4.8: Prikaz jakosti električne struje punjača prije punjenja.

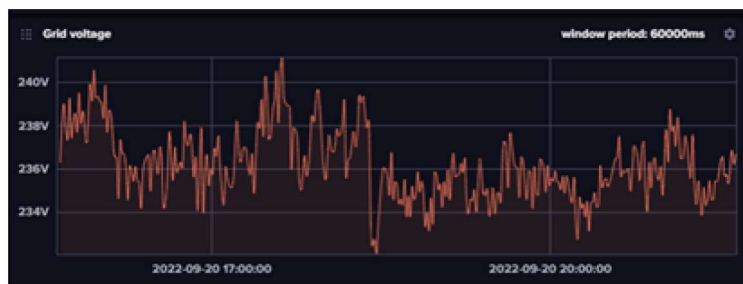


Slika 4.9: Prikaz napona punjača tijekom punjenja.

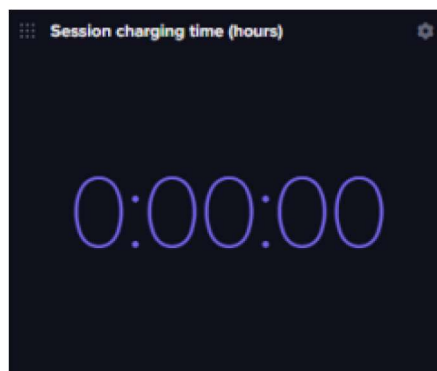


Slika 4.10: Prikaz napona punjača prije punjenja.

Ćelija **Total charging time** predstavlja ukupno vrijeme koliko je punjač punio automobil otkad je prvi put punjač uključen (uračunato je samo vrijeme u kojem je punjač punio automobil). Sagrađena je od podatka „charging_time_h” čija je vrijednost iskazana u formatu „dani, sati:minute:sekunde”. Slika 4.13 prikazuje ćeliju **Uptime** sagrađenu od podatka „uptime_h” koji ima vrijednost iskazanu u

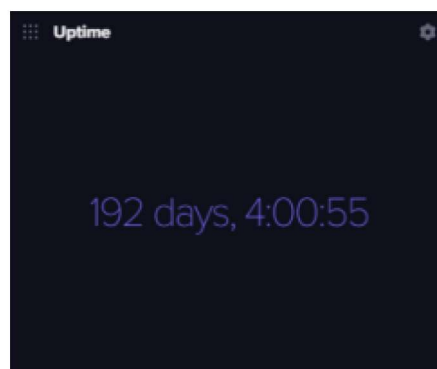


Slika 4.11: Prikaz napona mreže.



Slika 4.12: Prikaz da punjač nije uključen u automobil.

formatu „dani, sati:minute:sekunde”. Ta ćelija predstavlja ukupno vrijeme koliko je dugo punjač uključen u struju od kada je prvi put instaliran.



Slika 4.13: Ćelija Uptime.

Ćelija **Total energy consumption** predstavlja ukupnu količinu električne energije koju je punjač potrošio od početka svojeg rada. Sagrađena je od podatka „energy_kwh” čija je vrijednost iskazana u kWh mjernoj jedinici. Ćelija **Total electricity cost** je sagrađena od podatka „total_electricity_cost” koji ima vrijednost iskazanu u eurima, te predstavlja ukupnu cijenu potrošene električne energije od početka rada punjača po jeftinijoj tarifi. Slika 4.14 prikazuje ćeliju **Grid frequency**, koja predstavlja frekvenciju mreže, sagrađenu od podatka „grid_hz” čija je vrijednost iskazana u Hz mjernoj jedinici. Napon na zavojnici releja u punjaču predstavlja ćelija **Relay coil** sagrađena od podatka „relay_coil_v” čija je vrijednost

iskazana u voltima.



Slika 4.14: Prikaz frekvencije mreže.

Datoteka „tesla_wall_connector_gen_3.json“ je dashboard „Tesla Wall Connector Gen 3“ u JSON formatu. Poveznica za preuzimanje te datoteke s GitHuba je u poglavlju 5.2. Kako bi se ta datoteka umetnula u InfluxDB Cloud potrebno je najprije kreirati „bucket“ naziva „baza“.

5 | Implementacija

5.1 Organizacija koda

Ovaj projekt se sastoji od osam datoteka:

- `twc.py` je datoteka koja dohvaća podatke iz punjača i šalje ih u InfluxDB Cloud
- `ip_address.txt` je tekstualna datoteka koja sadrži IP adresu kućnog punjača
- `token.txt` je tekstualna datoteka koja sadrži token za autorizaciju na InfluxDB Cloud
- `url.txt` je tekstualna datoteka koja sadrži url za autorizaciju na InfluxDB Cloud
- `org.txt` je tekstualna datoteka koja sadrži org (naziv za radni prostor za grupu korisnika, tj. naziv organizacije) za autorizaciju na InfluxDB Cloud
- `tesla_wall_connector_gen_3.json` je dashboard u kojem se nalaze grafički prikazi prikupljenih podataka vremenskih nizova iz punjača u JSON formatu
- `start_twc_logger.sh` je datoteka koja omogućuje pokretanje novog terminala na Raspberry Pi u kojem se potom pokreće skripta `twc.py`
- `run-at-startup.service` je servis koji osigurava da se `start_twc_logger.sh` pokreće svaki put kada se Raspberry Pi pokrene.

U mapi naziva `twc` se nalaze datoteke `ip_address.txt`, `twc.py`, `url.txt`, `org.txt`, `token.txt` i `start_twc_logger.sh`. Ta mapa se nalazi na RPi mikro računalu koji će uporabom tih datoteka dohvaćati podatke, parsirati ih te neprekidno ih slati u InfluxDB Cloud u „bucket“ naziva „baza“.

5.2 GitHub repozitorij

Izvorni kod projekta se nalazi u GitHub repozitoriju:

<https://github.com/martina-os/Tesla-Wall-Connector-data-logger.git>

Literatura

- [1] *Chronograf*, URL: <https://medium.com/@MetricFire/what-is-chronograf-d0f3bfa6ccbe>
- [2] *COMPARISON, InfluxDB*, URL: <https://www.influxdata.com/lp/oss-vs-new-engine/>
- [3] *Get started with Flux*, URL: <https://docs.influxdata.com/flux/v0/get-started/>
- [4] *Get started with InfluxDB Cloud*, URL: <https://docs.influxdata.com/influxdb/cloud/get-started/>
- [5] *Grafana*, URL: <https://grafana.com/docs/grafana/latest/datasources/influxdb/>
- [6] *InfluxData: Products documentation*, URL: <https://docs.influxdata.com/>
- [7] *InfluxDB data elements*, URL: <https://docs.influxdata.com/influxdb/cloud/reference/key-concepts/data-elements/>
- [8] *Manage API tokens*, URL: <https://docs.influxdata.com/influxdb/cloud/admin/tokens/>
- [9] *Manage buckets*, URL: <https://docs.influxdata.com/influxdb/cloud/admin/buckets/>
- [10] *Manage organizations*, URL: <https://docs.influxdata.com/influxdb/cloud/admin/organizations/>
- [11] *Python client library*, URL: <https://docs.influxdata.com/influxdb/cloud/api-guide/client-libraries/python/>
- [12] *Visualize data with the InfluxDB UI*, URL: <https://docs.influxdata.com/influxdb/cloud/visualize-data/>
- [13] *What is a time series database?*, URL: <https://www.influxdata.com/blog/TSDB-data-lakes-together/>
- [14] *What is InfluxDB?*, URL: <https://www.influxdata.com/>
- [15] *Why use InfluxDB?*, URL: <https://www.influxdata.com/use-cases/real-time-analytics/>

Sažetak

Ovaj rad opisuje vizualizaciju prikupljenih podataka o statusu punjača iz Tesla Wall Connector Gen 3 punjača za električne automobile koristeći InfluxDB platformu. Tesla Wall Connector data logger kontinuirano prikuplja podatke poput potrošnje električne energije tijekom sesije, cijene potrošene električne energije tijekom sesije, trajanja punjenja, jakosti struje itd., te omogućuje grafički pristup tim podacima u InfluxDB Cloudu. U ovom je radu također opisan InfluxDB koji omogućuje vizualizaciju i statistiku podataka.

Ključne riječi

InfluxDB, Tesla Wall Connector Gen 3, data logger, vizualizacija podataka, Python, Raspberry Pi, podaci vremenskih nizova

Data visualization using the InfluxDB on example of data from the Tesla Wall Connector

Summary

This paper describes the visualization of collected charger status data from the Tesla Wall Connector Gen 3 charger for electric cars using the InfluxDB platform. The Tesla Wall Connector data logger continuously collects data such as electrical energy consumption during session, price of electricity consumed during the session, charging duration, current strength, etc., and provides graphical access to this data in the InfluxDB Cloud. This paper also describes InfluxDB which enables data visualization and statistics.

Keywords

InfluxDB, Tesla Wall Connector Gen 3, data logger, data visualization, Python, Raspberry Pi, time series data

Životopis

Rođena sam 14.10.1998. u Osijeku. Pohađala sam Osnovnu školu Grigor Vitez u Osijeku. Godine 2013. se upisujem u Prirodoslovno-matematičku gimnaziju u Osijeku. 2017. godine završavam gimnaziju te iste godine upisujem sveučilišni prijediplomski studij Matematika i računarstvo na Odjelu za matematiku u Osijeku, sada Fakultet primijenjene matematike i informatike, koji završavam 2024. godine.