

Utjecaj ventilacije na mikroklimatske parametre Rudnika sv. Barbara

Šarko, David

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mining, Geology and Petroleum Engineering / Sveučilište u Zagrebu, Rudarsko-geološko-naftni fakultet**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:169:985315>

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

Download date / Datum preuzimanja: **2024-07-30**



Repository / Repozitorij:

[Faculty of Mining, Geology and Petroleum Engineering Repository, University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
RUDARSKO-GEOLOŠKO-NAFTNI FAKULTET
Diplomski studij Rudarstva

**UTJECAJ VENTILACIJE NA MIKROKLIMATSKE PARAMETRE RUDNIKA
SV. BARBARA**

Diplomski rad

David Šarko
R4168

Zagreb, 2022



KLASA: 602-01/22-01/205
URBROJ: 251-70-11-22-2
U Zagrebu, 04.11.2022.

David Šarko, student

RJEŠENJE O ODOBRENJU TEME

Na temelju vašeg zahtjeva primljenog pod KLASOM 602-01/22-01/205, URBROJ: 251-70-11-22-1 od 04.11.2022. priopćujemo vam temu diplomskog rada koja glasi:

UTJECAJ VENTILACIJE NA MIKROKLIMATSKE PARAMETRE RUDNIKA SV. BARBARA

Za voditelja ovog diplomskog rada imenuje se u smislu Pravilnika o izradi i obrani diplomskog rada Izv.prof.dr.sc. Mario Klanfar nastavnik Rudarsko-geološko-naftnog-fakulteta Sveučilišta u Zagrebu

Voditelj

(potpis)

Izv.prof.dr.sc. Mario Klanfar

(titula, ime i prezime)

Predsjednik povjerenstva za
završne i diplomske ispite:

(potpis)

Izv.prof.dr.sc. Mario Klanfar

(titula, ime i prezime)

Prodekan za nastavu i studente:

(potpis)

Izv.prof.dr.sc. Borivoje
Pašić

(titula, ime i prezime)

Utjecaj ventilacije na mikroklimatske parametre rudnika Sv.Barbara

David Šarko

Rad izrađen: Sveučilište u Zagrebu
Rudarsko-geološko-naftni fakultet
Zavod za rudarstvo i geotehniku
Pierottijeva 6, 10 000 Zagreb

Sažetak

U ovom diplomskom radu analizirat će se podatci prikupljeni mrežom senzora koji su postavljeni unutar rudnika koristeći programski jezik Python. Na temelju podataka prikazani su dijagrami i histogrami koji će pokazati uspješnost rada novog ventilatora koji je ugrađen radi smanjivanje vlage unutar rudnika. Analizom podataka koji su učitani u python i potom uz korištenje programskog koda i algoritama dobiveni su rezultati kretanja rosišta, temperature i relativne vlage kroz vrijeme. Dobivena su ukupna vremena kad je ventilator radio tlačno, depresijski i kad je bio ugašen. Pokazano je da je ventilator opravdan u rudniku jer postoji nepovoljno vrijeme gdje bi moglo doći do kondenzacije vlage ako se ne upali ventilator. Koristeći unutrašnje senzore pokazalo se da nikad nije došlo do kondenzacije vlage unutar rudnika. Prikazana su kretanja apsolutne vlage kad ventilator radi i kad je ugašen. Analizom je prikazano da apsolutna vlaga pada dok ventilator radi, a raste kad je ugašen. Također u različitim periodima godine postoje i slučajevi gdje apsolutna vlažnost pada na sensorima bliže ulazu zračne struje a poraste na sensorima bliže izlazu zračne struje.

Ključne riječi: Relativna vlaga, apsolutna vlaga, temperatura, ventilator, rosište, senzor, Python

Diplomski rad sadrži: 52 stranice, 2 tablica, 23 slika, 3 priloga, i 14 reference.

Jezik izvornika: Hrvatski

Pohrana rada: Knjižnica Rudarsko-geološko-naftnog fakulteta, Pierottijeva 6, Zagreb

Mentori: Dr.sc. Mario Klanfar, izvanredni profesor RGNF

Ocjenjivači: Dr.sc. Mario Klanfar, izvanredni profesor RGNF
Dr.sc. Dubravko Domitrović, izvanredni profesor RGNF
Dr.sc. Tomislav Korman, izvanredni profesor RGNF

Influence of ventilation on the microclimate parameters of the mine of St. Barbara

David Šarko

Thesis completed at: University of Zagreb
Faculty of mining, Geology and Petroleum Engineering
Department of mining and geotechnics
Pierottijeva 6, 10 000 Zagreb

Abstract

This thesis will analyze data collected by a network of sensors placed inside the mine using the Python programming language. Based on the data, diagrams and histograms are presented that will show the success of the new fan installed to reduce moisture inside the mine. By analyzing the data that was loaded into python and then with the use of program code and algorithms, the results of the movement of the dew point, temperature and relative humidity over time were obtained. The total times when the ventilator was working in push mode, in pull mode and when it was turned off were obtained. It has been shown that the fan is justified in the mine because there is unfavorable weather where moisture condensation could occur if the fan is not turned on. Using internal sensors, it was shown that moisture condensation never occurred inside the mine. The movements of absolute humidity are shown when the fan is running and when it is turned off. The analysis showed that the absolute humidity falls while the fan is running, and rises when it is turned off. Also, in different periods of the year, there are cases where the absolute humidity drops on the sensors closer to the entrance of the air stream and rises on the sensors closer to the exit of the air stream.

Keywords: Relative humidity, absolute humidity, temperature, fan, dew point, Python

Thesis contains: 52 pages, 2 tables, 23 figures, 3 appendixes, i 14 references.

Original in: Croatian

Archived in: Library of Faculty of Mining, Geology and Petroleum Engineering, Pierottijeva 6, Zagreb

Supervisors: Associate Professor Mario Klanfar, PhD

Reviewers: Associate Professor Mario Klanfar, PhD
Associate Professor Dubravko Domitrović, PhD
Associate Professor Tomislav Korman, PhD

SADRŽAJ

1. UVOD	1
2. RUDNIK SV. BARBARA	2
3. KLIMATSKI PARAMETRI	4
3.1. Vlažnost zraka	4
3.2. Temperatura rosišta	5
4. VJETRENJE	8
4.1. Vrste vjetrenja	9
4.1.1. Ventilator	11
5. PROGRAMSKI JEZIK PYTHON	12
5.1. Radno okruženje	13
5.2. Osnove Pythona	14
5.2.1. Varijable	14
5.2.2. Aritmetički operatori	15
5.2.3. Operatori Usporedbe	15
6. ANALIZA PODATAKA	16
6.1. Senzori	18
7. REZULTATI OBRADE U PYTHONU	20
7.1. Učitavanje txt.datoteke u Python	20
7.2. Kretanje temperature, relativne vlage i rosišta kroz godinu	22
7.2.1. Povoljno i nepovoljno rosište	24
7.3. Rad ventilatora	28
7.3.1. Utjecaj Ventilacije na apsolutnu vlagu unutar rudnika	31
8. ZAKLJUČAK	35
9. LITERATURA	36

POPIS SLIKA

Slika 2-1. Lokacija rudnika.....	2
Slika 2-2. Ulazak u Rudnik.....	3
Slika 2-3. Stara oktopna radilišta u sideritu.....	3
Slika 3-1. Količina vodene pare u zraku.....	5
Slika 3-2. Psihometrijska svojstva vlažnog zraka.....	7
Slika 4-1. Sustavi vjetrenja.....	9
Slika 4-2. Strujanje zraka pri različitim vremenskim dobima.....	10
Slika 4-3. Odabrani ventilator u rudniku Sv.Barbare.....	11
Slika 5-1. Python logo.....	12
Slika 5-2. Python sučelje IDLE.....	14
Slika 6-1. kontrolna ploča.....	16
Slika 6-2. Izgled podataka koji bilježe senzori.....	17
Slika 6-3 Vanjski senzor Rotronic HF1.....	18
Slika 6-4 Unutarnji senzor.....	19
Slika 7-1. Struktura podataka.....	21
Slika 7-2. Povoljno i nepovoljno vrijeme.....	24
Slika 7-3. Povoljno/ nepovoljno kroz mjesece.....	26
Slika 7-4. povoljno/nepovoljno unutarnji senzori.....	27
Slika 7-5. Rad ventilatora.....	28
Slika 7-6 Utjecaj ventilacije na apsolutnu vlagu unutar rudnika (idealni slučaj).....	32
Slika 7-7. Utjecaj ventilacije na apsolutnu vlagu unutar rudnika (karakteristični intervali 1).....	33
Slika 7-8 Utjecaj ventilacije na apsolutnu vlagu unutar rudnika (karakteristični intervali 2).....	34

POPIS TABLICA

Tablica 5-1. aritmetički operatori.....	15
Tablica 5-2. operator usporedbe.....	15

POPIS PRILOGA

Prilog 1. Karta vjetrenja sa lokacijama senzora.....	37
Prilog 2. Temperatura, relativna vlažnost i rosište.....	38
Prilog 3. Programski kod u Pythonu.....	39-52

POPIS KORIŠTENIH OZNAKA I JEDINICA

Oznaka	Jedinica	Opis
X	g/cm ³	Apsolutna vlažnost zraka
t	°	temperatura zraka
P	Pa	tlak zraka
φ	%	relativa vlažnost zraka
Tr	°C	temperatura rosišta
e	Pa	parcijalni tlak vodene pare

1. UVOD

Rudnik Sv Barbare nalazi se u mjestu Rude, pet kilometara jugozapadno od Samobora. O bogatoj rudarskoj tradiciji Ruda svjedoče i sama imena mjesta koja su se u povijesti mijenjala, ali je njihovo značenje uvijek bilo povezano s rudom, primjerice Rovi i Fodinae. Nekada je kroz cijelo mjesto bila razvedena mreža rudarskih rovova, no danas je za posjetitelje obnovljeno 350 metara rovova umjetnim spajanjem rovova Sveto Trojstvo i Kokel. Tijekom 20. st. provedena su mnoga istraživanja čiji je cilj bio ponovno oživljavanje rudarstva u Rudama, no ona su pokazala da bakrene i željezne rude nema u količinama koje bi bile ekonomične za iskapanje. Od tada rudarenje kao način života u Rudama zamire. Godine 2002. započela je obnova i turističko oživljavanje nekadašnjih rovova. Dijelovi napuštenog rudnika pretvoreni su u rudarski muzej u prirodi i time je sačuvan uvid u ljepotu podzemlja Ruda, a dragocjenu baštinu vezanu uz težak rudarski život spašen je od zaborava. (KUD Oštrc, 2016).

Prije nekoliko godina u suradnji KUD Oštrca i Rudarsko-geološko-naftnog fakulteta ugrađen je novi ventilator promjenjivog smjera vrtnje i bitno većeg kapaciteta od postojećeg . Promjenom smjera vrtnje dobivena je mogućnost upuhivanja ili izvlačenja zraka iz rudarskog niskopa, ovisno o vanjskoj temperaturi. Automatski sustav kontrole i regulacije vlage po potrebi će uključivati i isključivati ventilator, što bi prema očekivanjima trebalo imati za posljedicu smanjenje vlage u rudniku i usporavanje procesa truljenja drvene podgrade. Krajem 2020. kroz cijeli rudnik postavljena je mreža senzora koja kontinuirano bilježi temperaturu i vlagu. U ovom diplomskom radu analizirat će se podaci zabilježeni sensorima koristeći programski jezik Python. Riječ je o velikim količinama podataka (senzori bilježe podatke o temperaturi, vlazi i tlaku svaku minutu) tako da je analiza uz pomoć programskog jezika logičan izbor. Analiza će pokazati kakav je utjecaj ventilacije na smanjenje vlažnosti u rudniku, kakva je vlaga u rudniku (je li zadovoljavajuća ili ne), koliko efikasno novi sustav ventilacije radi i da li postoji potreba za optimizacijom rada istog.

2. RUDNIK SV. BARBARA

Rudnik sveta Barbara nalazi se Rudama, mjestu udaljenom 5 km jugozapadno od Samobora te na početku Samoborskog gorja (slika 2-1). Rudnik počinje na 300-injak metara nadmorske visine. Rudnik je obnovljen, a dio je cjelovitoga projekta pod nazivom „Bogatstvo podzemnog svijeta nekad i danas”. Rudarskim radovima od 2007. do početka 2012. godine obnovljeno je ukupno 350 metara rovova te je ispred njih postavljena info kućica (slika 2-2). Uređena je rudarsko-botanička poučna staza dugačka 1.500 metara, koja počinje kod OŠ Rude, a završava u Botaničkom vrtu Suban u Manjoj Vasi. Duž poučne staze postavljeno je 8 informativnih i interpretacijskih ploča te dvadesetak tabla s botaničkim nazivljem biljaka. Također, u Osnovnoj školi Rude uređena je vitrina sa zbirkom od 120 različitih uzoraka minerala, stijena i fosila, koja se stalno nadopunjuje novim uzorcima. Rudnik sveta Barbara sastoji se od tri rova: Vlašić, Sveto Trojstvo i Kokel. Za posjet su otvoreni dijelovi rovova Sveto Trojstvo i Kokel koji su povezani starim otkopima željezne rude (siderita) (slika 2-3), dok je rov Vlašić, nekada glavni izvozni rov bakrene rude, i dalje nedostupan. (Turistička zajednica Zagrebačke županije, 2022). Dakle, rudnik je danas otvoren isključivo u edukacijsko-turističke svrhe, ne postoje aktivni rudarski radovi unutar rudnika.



Slika 2-1. Lokacija rudnika



Slika 2-2. Ulazak u rudnik



Slika 2-3. Stara otkopna radilišta u sideritu

3. KLIMATSKI PARAMETRI

Pod klimatske parametre koji utječu na rad ventilatora podrazumijevamo tlak, temperaturu zraka, vlažnost zraka (koja može biti apsolutna ili relativna) te rosište, odnosno temperatura rosišta. Vlažnost i temperaturu bilježi vanjski senzor koji je postavljen u blizini ulaska u rudnik. Ostali senzori koji se nalaze unutar rudnika, osim vlažnosti i temperature bilježe i vrijednost tlaka. Temperatura rosišta se ne može dobiti očitanjem senzora, već nju dobivamo računski iz temperature zraka i relativne vlažnosti zraka. Najbitnije stavke o svim navedenim klimatskim parametrima i dobivanju temperature rosišta biti će spomenuti u sljedećim podpoglavljima.

3.1. Vlažnost zraka

Vlažnost zraka predstavlja količinu vodene pare u zraku pri određenom tlaku i temperaturi. Razlikujemo apsolutnu vlažnost i relativnu vlažnost. Apsolutna vlažnost predstavlja masu vodene pare u masi suhog zraka:

$$X = \frac{mv}{ms} \quad (3-1)$$

gdje je :

- mv – masa suhog zraka (kg)
- ms – masa vodene pare (kg)

Budući da je masa vodene pare u 1 kg suhog zraka relativno mala, apsolutna vlažnost X često se deklarira u gramima po 1 kg suhog zraka. Apsolutna vlažnost zraka u prirodi je između 2 i 20 g/kg, a apsolutna vlažnost zraka unutarnjih prostora je obično između 4 i 12 g/kg (Medved, 2021).

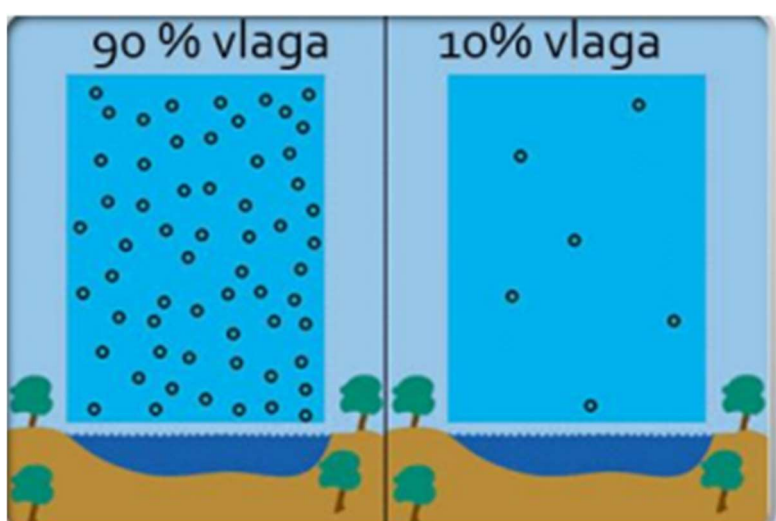
Relativna vlažnost predstavlja odnos trenutno izmjerenog sadržaja vodene pare u zraku i maksimalno mogućeg (100 % -tna zasićenost) (slika 3-1) :

$$\varphi = \frac{e}{E} * 100\% \quad (3-2)$$

Gdje je :

- e – parcijalni pritisak vodene pare u zraku (Pa)
- E – parcijalni pritisak vode u zasićenom zraku (Pa)

Prema Daltonovom zakonu, koji govori o tlaku smjese plinova u zatvorenoj posudi; ukupni tlak smjese idealnih plinova (koji međusobno ne reagiraju) jednak je zbroju djelomičnih (parcijalnih) tlakova. (Vrkljan, 2007). Stoga parcijalni tlak vodene pare (e) predstavlja tlak, koji bi razvijale molekule vode, kada bi bile "same" u atmosferi (bez molekula dušika, kisika, ugljičnog dioksida, plemenitih plinova).



Slika 3-1. Količina vodene pare u zraku. (Klanfar, 2018)

3.2. Temperatura rosišta

Rosište je temperatura pri kojoj dolazi do kondenzacije vlage u zraku. To je temperatura pri kojoj uz nepromijenjeni tlak i nepromijenjenu količinu vodene pare, zrak postaje zasićen vodenom parom (Hrvatska enciklopedija, 2020). Zrak je „primio“ maksimalnu količinu vodene pare, što znači da ima 100% relativnu vlažnost. Zbog toga višak vodene pare se kondenzira i manifestira se u obliku rose (tekuće agregatno stanje) ili mraza (čvrsto agregatno stanje). Snižavanjem temperature zraka smanjuje se tlak zasićene vodene pare, a do kondenzacije dolazi kada se parcijalni tlak vodene pare i tlak zasićene vodene pare izjednače. $E=e$, što znači da je relativna vlaga, $\varphi = 100\%$ (Šarko, 2020). Tu pojavu „skupljanja vlage“, odnosno kondenzaciju vodene pare želimo izbjeći jer ona direktno utječe na truljenje drvene podgrade u rudniku. Visok udio vlage i nakupljanje rose predstavlja optimalno mjesto za nastanak bakterija koje nepovoljno utječu na drvenu podgradu uzrokuju

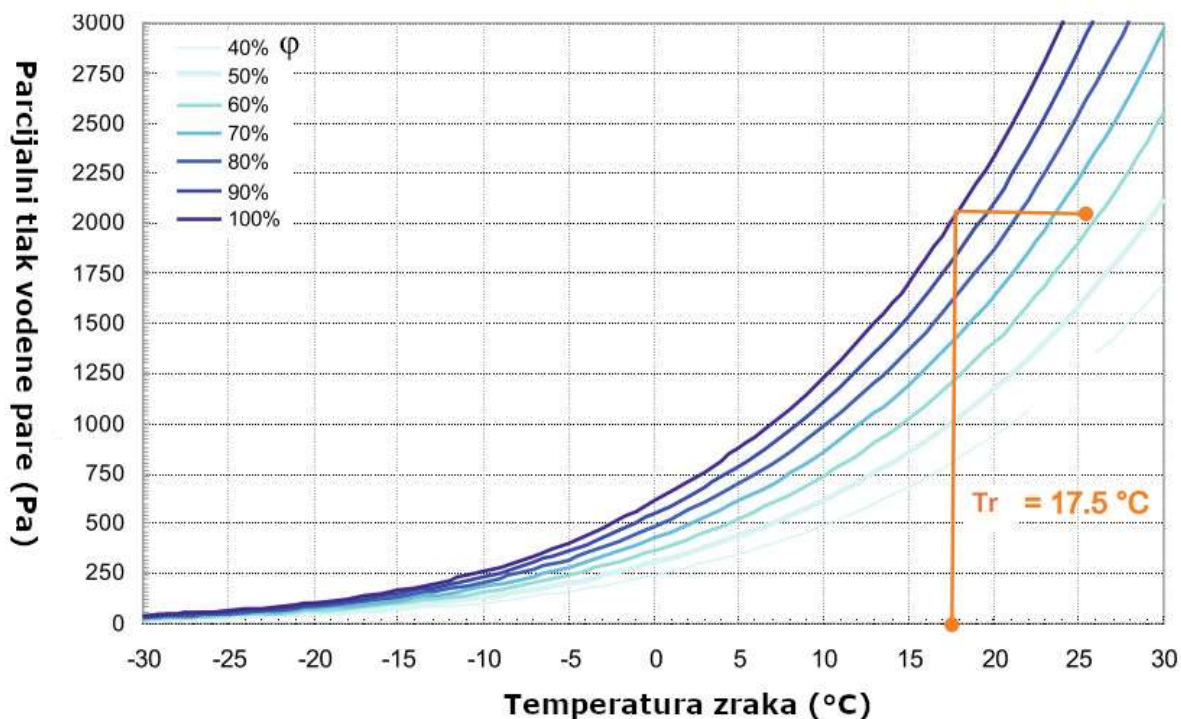
njezinu degradaciju. Time se narušava stabilnost i sigurnost u rudniku. To je primarni razlog instalacije novog sustava ventilacije. Temperatura rosišta unutar rudnika ne bi trebala prelaziti preko 12,5 °C koja predstavlja prosječnu temperaturu u rudniku, no detaljnije o tome će biti rečeno u poglavljima analize podataka senzora. Rosište se može odrediti na više načina. Nama najzanimljiviji su oni koji omogućavaju izračun temperature rosišta samo preko dva klimatska parametra, a to su temperatura zraka i relativna vlaga, upravo parametri koji bilježi mreža senzora postavljena unutar rudnika. Pojednostavljenu formulu za izračun temperature rosišta koja aproksimira temperaturu rosišta predložio je Mark G. Lawrence (2005).:

$$Tr = T - ((100 - RH)/5) \quad (3-3)$$

Gdje je :

- Tr – temperatura rosišta (°C)
- T- temperatura zraka (°C)
- RH – relativna vlažnost (°C)

Točnost ovog izraza je ± 1 °C, ako je relativna vlažnost iznad 50%, za manje vrijednosti relativne vlage izraz je manje točnosti. S obzirom da je relativna vlažnost unutar rudnika uvijek iznad 50% u proračunu ćemo koristiti navedenu formulu. Odnos između temperature , relativne vlažnosti, parcijalnog tlaka vodene pare te određivanje temperature rosišta pri određenim parametrima možemo vidjeti na slici 3-2. Pomoću dijagrama možemo odrediti temperaturu rosišta za određenu vrijednost temperature i relativne vlage.



Slika 3-2. Psihrometrijska svojstva vlažnog zraka (Medved, 2021)

Prikazane su različite vrijednosti relativne vlage, u razmaku od 20 pa sve do 100 % vlage koja predstavlja potpuno zasićenje vodenom parom te dolaskom do kondenzacije. U primjeru je dana relativna vlaga od 62 % te temperatura zraka od 24 °C. Na dijagramu nađemo sjecište tih dvaju vrijednosti te od sjecišta povučemo horizontalu do vrijednosti 100 % relativne vlage te na kraju od toga vertikalnu do x osi gdje dobivamo vrijednost temperature rosišta koju vidimo na slici. To je jedan od načina određivanja temperature rosišta, pomoću dijagrama, ali opet zapazimo kako su za određivanje rosišta i u ovom slučaju trebali samo podaci o relativnoj vlazi i temperaturi zraka.

4. VJETRENJE

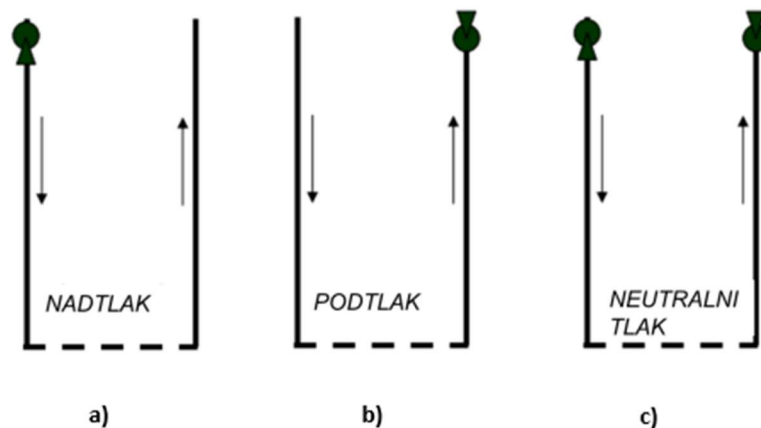
Za rudarskog inženjera, vjetrenje je najsvestraniji alat za kontrolu atmosfere. Vjetrenje rudnika je u biti primjena načela dinamike fluida na strujanje zraka u rudničkim otvorima. Vjetrenje se primarno koristi kao sredstvo kontrole količine zraka u rudniku, odnosno za unos svježeg zraka izvana kako bi radnicima osigurali uvjete za rad. Druga bitna stavka zbog koje se koristi vjetrenje jest kontrola plinova unutar rudnika. Tri najbitnije stavke korištenja vjetrenja (Hartman i dr., 1997) :

1. Kontrola kvalitete (pročišćavanje zraka i uklanjanje onečišćenja)
 - a. Kontrola plinova—pare i plinovite tvari, uključujući zračenje
 - b. Kontrola prašine—čestice
2. Kontrola količine (regulacija veličine i smjera strujanja zraka)
 - a. Ventilacija
 - b. Seperatna ventilacija
 - c. Lokalna ispušna ventilacija
3. Kontrola temperature i vlage (kontrola latentne i osjetljive topline)
 - a. Hlađenje
 - b. Grijanje
 - c. Ovlaživanje
 - d. Odvlaživanje

Međutim, s obzirom da rudnik Sv. Barbare nije aktivan rudnik, odnosno u njemu se ne eksploatira ruda, možemo zanemariti prve dvije stavke. U našem slučaju, ventilaciju koristimo za kontrolu temperature i vlage kako bi usporjeli proces truljenja drvene podgrade u rudniku. Najčešće se to odnosi na odvlaživanje zraka, ne smijemo dopustiti nakupljanje prevelike količine vlage te njezino kondenziranje unutar rudnika jer bi to imalo negativni utjecaj na podgradu.

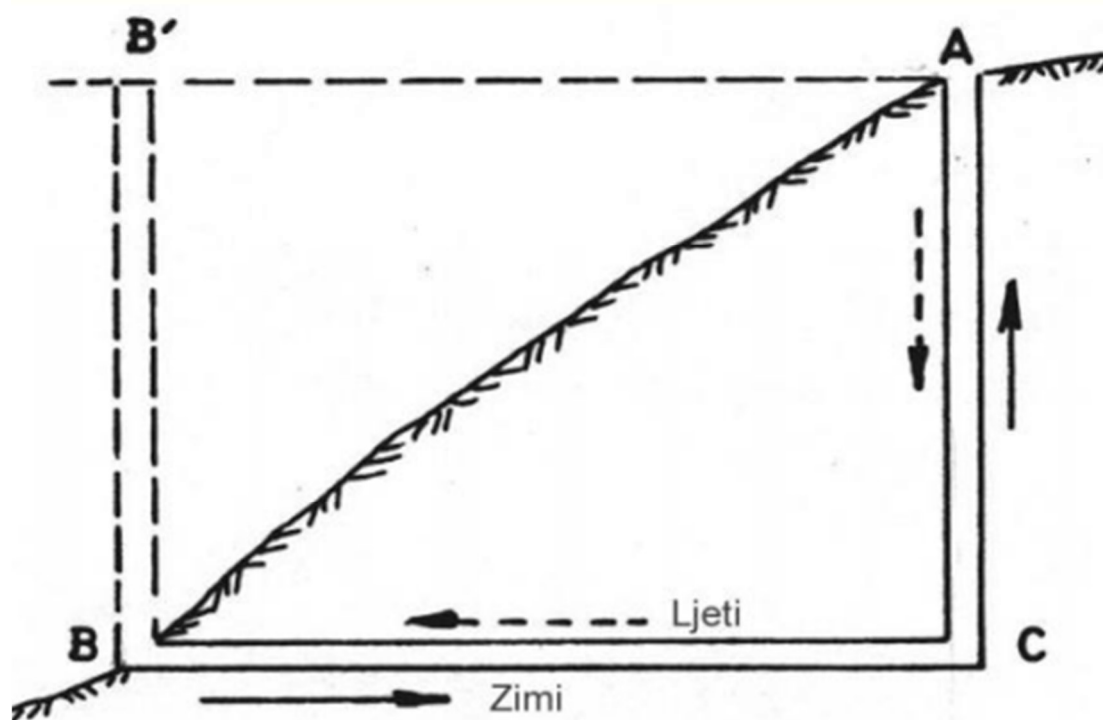
4.1. Vrste vjetrenja

Vjetrenje se može postići na nekoliko načina. Najjednostavnija te ujedino i najjeftinija metoda je vjetrenje prirodnom depresijom. Ako je velika visinska razlika između izlaza i ulaza u rudnik postoji mogućnost vjetrenja isključivo preko prirodne depresije. Zbog razlike potencijala između izlaza i ulaza do vjetrenja dolazi prirodnim putem i ponekad je to dovoljno za prihvatljivo strujanje zraka u rudniku. Primjer takvog vjetrenja možemo pronaći u rudnicima boskita u Jajcu. Međutim, u današnjim, suvremenim rudnicima jedini način rješavanja ventilacijskih i klimatizacijskih problema jame predstavlja mehanička ventilacija, a prirodna (toplinska) depresija je samo jedan od faktora koji u pozitivnom ili negativnom smislu utječe na tu mehaničku ventilaciju. Nekada je prirodna depresija bila jedini način rješavanja ovih problema. Da bi se postigao potreban protok zraka u jami nekadašnji rudari ložili su vatre, koristili su cijedenje vode niz ulazno vjetreno okno i vjetrene pregrade (Vrkljan, 2007). Kao pogonska sila kod mehaničkog sistema ventilacije djeluju ventilatori. Postoje različiti načini djelovanja ventilatora: tlačni ili kompresijski, usisni ili depresijski te Balansirani ili push-pull sustav. Svaki sustav ima svoje prednosti i nedostatke. Tlačni sustav (slika 4-1, a) tlači zrak u ležište i pri tome dolazi do nadtlaka u ležištu. Kod takvog sustava transport se odvija u izlaznoj struji, loša kontrola plinova ali manji su troškovi održavanja jer čisti vanjski zrak prolazi kroz ventilator. Depresijski sustav (slika 4-1, b) radi obrnuto od tlačnog, on vuče plinove i zrak iz ležišta u jednu točku. Prednost je što se transport odvija u ulaznoj svježoj zračnoj struji, no veći su troškovi održavanja zbog prašine, vlage i plinova koji prolaze kroz ventilator. Balansirani sustav (slika 4-1, c) kombinira prednosti tlačnog i depresijskog sustava, tlači se zrak u ležište, tlak je izjednačen s površinom te su najmanji gubici prema površini (Klanfar, 2018).



Slika 4-1. Sustavi vjetrenja (Klanfar, 2018)

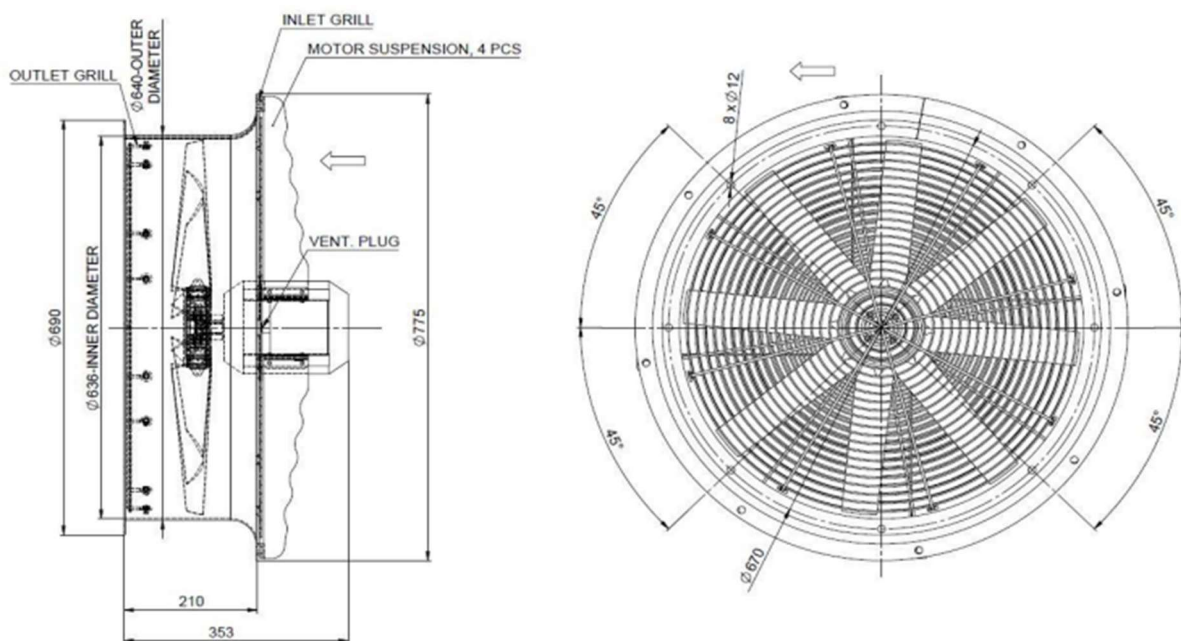
Pod pojmom depresije označava se u načelu promjena tlaka u određenoj dionici zračnog provodnika (hodnika). Ta promjena očituje se padom tlaka, izazvanim otporom strujanja. Do pada, odnosno gubitka tlaka dolazi zbog promjene smjera zračne struje. Promjena zračne struje, odnosno strujanje zraka uopće nastaje pod utjecajem razlike u pritiscima (slika 4-2) dva zračna stupca- vanjskog BB' i janskog AC. Naime, zimi, kada je zrak na površini hladan, ima veću specifičnu težinu od toplijeg zraka u rudniku, pa će nastati strujanje zraka kroz potkop i na okno izvan rudnika. Ljeti kada je atmosferski zrak topliji, biti će i specifična težina tog zračnog stupca manja od one u oknu BC, budući da je tamo sada temperatura niža, pa će i strujanje zraka biti obrnuto : zrak će ulaziti kroz okno, a izlaziti na potkop (Vrkljan, 2007). Poznavanje kretanja zraka pri različitim vremenskim dobima bitna je stavka pri odabiru ventilatora i njegovom načinu rada.



Slika 4-2- Strujanje zraka pri različitim vremenskim dobima (Klanfar, 2018)

4.1.1. Ventilator

Ventilatori su uređaji koji proizvode mehaničku energiju i predaju je fluidu (zraku), ostvarujući pri tome povećanje tlaka i protok fluida. Ventilatori se suprotstavljaju otporima strujanju (gubicima energije) i održavaju ravnotežni tlak pri strujanju zrak (Klanfar, 2018). Postoje različite izvedbe ventilatora, najpoznatiju su centrifugalni, aksijalni te poprečno protočni. Odabrani ventilator (slika 4-3) u rudniku Sv. Barbare je prema izvedbi aksijalni ventilator. Zrak u cilindrično kućište ulazi aksijalno i izlazi van ventilatora aksijalno. Lopatice daju ubrzanje zraku okomito na smjer rotacije(Klanfar, 2018). S obzirom kako je primarni cilj ventilatora u rudniku Sv. Barbare sprječavanje nakupljanje vlage unutar rudnika, odabran je reverzibilni ventilator. To znači da ventilator može raditi tlačnim i depresijskim načinom. Kao što je prije navedeno, s obzirom da postoji prirodna depresija koja uzrokuje promjene strujanja zraka unutar rudnika, reverzibilni način rada ventilatora omogućuje da ventilator nikad ne radi u smjeru obrnutom od prirodnog strujanja zraka unutar rudnika, prirodna depresija mu služi samo u pozitivnom smislu.



Slika 4-3. Odabrani ventilator u rudniku Sv. Barbare

Ventilator može postići 1360 okretaja po minuti, mogućnost rada na temperaturama od -20° C do 55° C. Ventilator ima električni motor snage 1.1 kW.

5. PROGRAMSKI JEZIK PYTHON

Python je objektno orijentirani jezik koji je razvijen kasnih 1980-ih kao skriptni jezik (naziv je izveden iz britanske televizijske emisije “Monty Pythonov leteći cirkus“). Iako Python nije toliko poznat u inženjerskim krugovima kao neki drugi jezici, ima znatnu popularnost kod zajednice programera. U stvari, Python (slika 5-1) je znatno rašireniji od Fortran-a. Python se može promatrati kao jezik u nastajanju, budući da se još uvijek razvija i usavršava. U trenutnom stanju, to je izvrstan jezik za razvoj inženjerskih aplikacija i rješavanja problema obrade podataka jer posjeduje jednostavnu eleganciju s kojom se drugi programski jezici ne mogu mjeriti (Kiusalaas, 2010). Python programi nisu kompajlirani u strojni kod, već ih pokreće interpreter. Velika prednost interpretiranog jezika je u tome što se programi mogu testirati i brzo se otklanjaju pogreške, dopuštajući korisniku da se više koncentrira na principe iza programa, a manje na samo programiranje. Budući da nema potrebe za kompajliranjem, povezivanjem i izvršavanjem nakon svake korekcije, Python programi se mogu razviti u mnogo kraće vrijeme od ekvivalentnih Fortran ili C programa. S negativne strane, intepretirani programi ne proizvode samostalne aplikacije. Tako Python program može biti pokrenuti samo na računalima koja imaju instaliran Python interpreter. Python ima druge prednosti u odnosu na „mainstream“ jezike koje su važne u okolini za učenje (Kiusalaas, 2010):

- Python je softver otvorenog koda, što znači da je besplatan; uključen je u većinu Linux distribucije
- Python je dostupan za sve glavne operativne sustave (Linux, Unix, Windows, Mac Os itd.). Program napisan na jednom sustavu radi bez izmjena na svim sustavima.
- Python je lakši za naučiti i proizvodi čitljiviji kod nego drugi jezici
- Python i njegova proširenja lako se instaliraju



Slika 5-1. Python logo (izvor : <https://en.wikiversity.org/wiki/Python>)

5.1. Radno okruženje

Nabavljanje i instalacija pythona je vrlo jednostavna. Na stranici <https://www.python.org/> odaberemo zadnju verziju pythona (u našem slučaju Python 3.10). Prilikom instalacije možemo koristiti unaprijed postavljene opcije tj. nije potrebno ništa mijenjati. Kad smo to napravili možemo otvoriti radni prostor u kojem pišemo programski kod, on se zove IDLE (slika 5-2).

Start → Python 3.10 → IDLE (Python 3.10 64-bit)

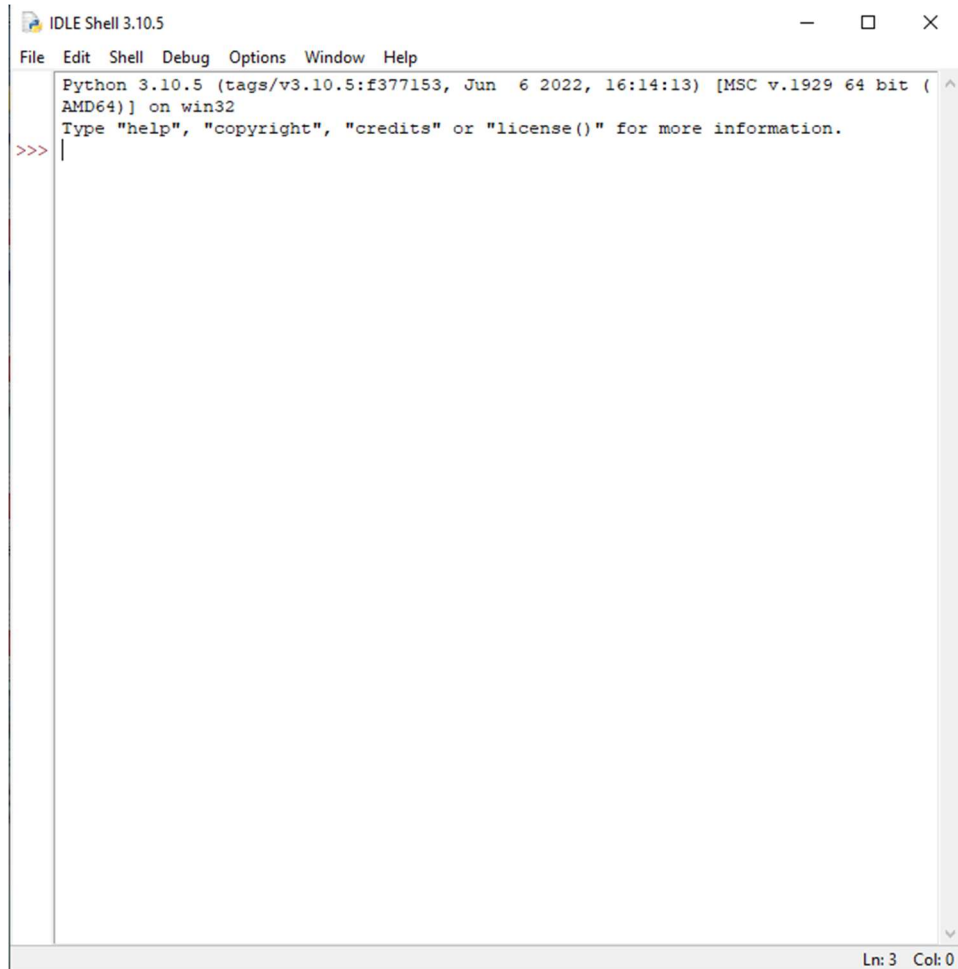
Kod ovakvog pristupa programiranju, program koji napišemo je "jednokratani" što znači da ako ponovo želimo pokrenuti isti skup programskih linija, sve te linije moramo ponovo napisati ili koristiti metodu kopiraj-zalijepi (engl. Copy-Paste). Ako želimo napisati program koji se sastoji od više linija, koji lagano možemo uređivati kao i svaki drugi dokument na računalu, za to je najprikladnije da preko Python IDLE-a otvorimo novi dokument u kojem ćemo pisati programski kod: U programu koji smo prethodno otvorili: IDLE (Python 3.10 64-bit) odaberemo:

File → New file

Primijetimo da nam se sada otvorio prozor za unos teksta. U ovom prozoru pišemo programski kôd. Nakon što želimo pokrenuti napisani program to možemo napraviti pritiskom na tipku: F5 ili pak kroz izbornik:

Run → Run module

Nakon što pokrenemo napisani programski kod, rezultat izvođenja programa pojaviti će nam se u prozoru koji smo prvo otvorili, tj. IDLE (Python 3.10 64-bit). (Hruška, 2018).



Slika 5-2. Python sučelje IDLE

5.2. Osnove Pythona

Kako bismo se znali služiti programskim jezikom python, trebamo znati njegov „jezik“ odnosno kako pisati u njemu i koje sve mogućnosti ima. Trebamo znati razlikovati varijable, logičke operatore, tipove brojeva itd.

5.2.1. Varijable

Varijable su dijelovi programa koji se koriste kako bi se spremile vrijednosti. Tim vrijednostima (informacijama) pristupa se iz raznih dijelova računalnih programa. Svaka varijabla određena je svojim imenom i memorijskom lokacijom na kojoj je zapisana određena vrijednost. Najjednostavniji opis pojma varijabla bio bi da je varijabla "kutija" u koju se spremaju stvari (u našem slučaju vrijednosti) te se prema potrebi te stvari mogu uzeti (koristiti, pridruživati drugim varijablama), mijenjati s drugim stvarima (s drugim vrijednostima) itd. spremati samo brojevi radi lakšeg razumijevanja gradiva. Vrijednosti se

nekoj varijabli pridružuju pomoću matematičkog operatora "=". Ime varijable u koju želimo spremi neku vrijednost uvijek se nalazi s lijeve strane znaka "=" dok se s desne strane nalazi vrijednost koju želimo spremi u varijablu (Hruška, 2018). Kad kreiramo varijablu, preporučljivo je nazvati ju na prikladan način kako kasnije ne bi zaboravili što ona znači, odnosno da si olakšamo snalaženje u programu. Primjer jedne varijable :

```

TemperaturaSenzorSHZ1 = 20
print(TemperaturaSenzorSHZ1)
    izlaz :
        20

```

5.2.2. Aritmetički operatori

Python podržava nekoliko unaprijed definiranih aritmetičkih operatora za rad s brojevima. U tablici 5-1 se nalazi popis aritmetičkih operatora koji se mogu koristiti.

Tablica 5-1. aritmetički operatori (Hruška, 2018)

Operator	Namjena	Primjer	Rezultat a = 10; b = 5
+	Zbrajanje (engl. <i>addition</i>)	x = a + b	x = 15
-	Oduzimanje (engl. <i>subtraction</i>)	x = a - b	x = 5
*	Množenje (engl. <i>multiplication</i>)	x = a * b	x = 50
/	Dijeljenje (engl. <i>division</i>)	x = a / b	x = 2
%	Ostatak dijeljenja (engl. <i>modulus</i>)	x = a % b	x = 0
**	Potenciranje (engl. <i>power</i>)	x = a ** b	x = 100000
//	Cjelobrojno dijeljenje (engl. <i>flor division</i>)	x = a // b	x = 2

5.2.3. Operatori Usporedbe

Operatori usporedbe uspoređuju vrijednosti s lijeve i s desne strane operatora te odlučuju o vrijednosti koju će vratiti. Operatori usporedbe vraćaju vrijednosti 0 (netočno, engl. False) ili 1 (točno, engl. True).

Tablica 5-2. operator usporedbe (Hruška, 2018)

Operator	Opis	Primjer	Rezultat a = 5; b = 10	Rezultat a = 5; b = 5
==	Usporedba jednakosti	a == b	False	True
!=	Usporedba nejednakosti	a != b	True	False
>	Strogo veće od	a > b	False	False
<	Strogo manje od	a < b	True	False
>=	Veće ili jednako od	a >= b	False	True
<=	Manje ili jednako od	a <= b	True	True

6. ANALIZA PODATAKA

U rudnik je postavljena mreža senzora koji bilježe podatke o temperaturi, relativnoj vlazi te tlaku unutar rudnika i neposredno prije ulaska u rudnik. Lokaciju senzora i tlocrt rudnika možemo vidjeti u prilogu 1. Navedeni parametri se bilježe svaku minutu kroz cijelu godinu i time čine gust i količinski velik skup podataka. Podaci se prikupljaju preko centralne ploče (slika 6-1) i spremaju se u tekstualni file (slika 6-2). Zbog velikih količina podataka postoje dvije datoteke, u jednoj su podaci od 11. mjeseca 2020 do 6. mjeseca 2021 koji je nazvan data 1, a u drugoj od 6. mjeseca 2021 do 11 mjeseca 2021 nazvan data 2.



Slika 6-1. kontrolna ploča

The screenshot shows a Notepad window titled 'Data_1_6.txt - Notepad'. The window contains a log of sensor data. Each entry consists of a timestamp, a sensor ID, a message type, and numerical data. The data is organized into columns: timestamp, sensor ID, message type, and three numerical values (or a status value).

Timestamp	Sensor ID	Message Type	Value 1	Value 2	Value 3
2020-10-30 12:55:46	PLC	MSG	PLC-STARTED		
2020-10-30 12:55:46	FAN	DATA	0		
2020-10-30 12:55:47	ADC0	DATA	12.66	84.16	
2020-10-30 12:56:42	SHZ1	DATA	-255.00	-255.00	990.30
2020-10-30 12:56:42	UPR2	DATA	14.69	85.75	989.00
2020-10-30 12:56:42	SPT3	DATA	14.61	90.76	985.55
2020-10-30 12:56:42	ERC4	DATA	16.03	84.47	986.92
2020-10-30 12:56:42	LOQ5	DATA	14.10	92.23	987.36
2020-10-30 12:56:42	YDW6	DATA			
2020-10-30 12:56:43	QZD7	DATA			
2020-10-30 12:56:43	HJP8	DATA	14.11	91.46	988.88
2020-10-30 12:56:44	ALK9	DATA			
2020-10-30 12:59:42	FAN	DATA	1		
2020-10-30 13:00:47	ADC0	DATA	12.77	83.99	
2020-10-30 13:01:12	RAO	DATA			
2020-10-30 13:01:42	SHZ1	DATA	-255.00	-255.00	990.34
2020-10-30 13:01:42	UPR2	DATA	14.65	85.93	989.00
2020-10-30 13:01:42	SPT3	DATA	14.60	90.75	985.44
2020-10-30 13:01:42	ERC4	DATA	16.02	84.46	986.94
2020-10-30 13:01:43	LOQ5	DATA	14.10	92.14	987.20
2020-10-30 13:01:43	YDW6	DATA			
2020-10-30 13:01:43	QZD7	DATA			
2020-10-30 13:01:44	HJP8	DATA	14.08	91.53	988.87
2020-10-30 13:01:44	ALK9	DATA			
2020-10-30 13:05:47	ADC0	DATA	12.80	84.82	
2020-10-30 13:06:42	SHZ1	DATA	-255.00	-255.00	990.30
2020-10-30 13:06:42	UPR2	DATA	14.66	85.85	988.86
2020-10-30 13:06:43	SPT3	DATA	14.60	90.78	985.39
2020-10-30 13:06:43	ERC4	DATA	16.03	84.40	986.87
2020-10-30 13:06:43	LOQ5	DATA	14.12	92.07	987.09
2020-10-30 13:06:43	YDW6	DATA			
2020-10-30 13:06:43	QZD7	DATA			
2020-10-30 13:06:44	HJP8	DATA	14.05	91.61	988.72
2020-10-30 13:06:44	ALK9	DATA			
2020-10-30 13:10:47	ADC0	DATA	12.79	85.10	
2020-10-30 13:11:12	RAO	DATA			
2020-10-30 13:11:42	SHZ1	DATA	-255.00	-255.00	990.41
2020-10-30 13:11:43	UPR2	DATA	14.65	85.93	988.83
2020-10-30 13:11:43	SPT3	DATA	14.61	90.77	985.41
2020-10-30 13:11:43	ERC4	DATA	16.02	84.45	986.77
2020-10-30 13:11:43	LOQ5	DATA	14.13	92.00	987.12
2020-10-30 13:11:43	YDW6	DATA			
2020-10-30 13:11:43	QZD7	DATA			
2020-10-30 13:11:44	HJP8	DATA	14.02	91.72	988.74
2020-10-30 13:11:44	ALK9	DATA			

Slika 6-2. Izgled podataka koji bilježe senzori

Na slici 6-2 možemo primijetiti da su podatci stavljeni u stupce i redke, što nam je kasnije bitno prilikom učitavanja podataka u Python. Prvi stupac označava trenutno vrijeme mjerenja, drugi senzor na kojemu je izvršeno očitavanje, a treći podatke o temperaturi, vlazi i tlaku. No ako je riječ o ventilatoru, a ne o senzoru onda je zapisano 0,1 ili -1, što označava mod ventilatora (ugašen, upaljen tlačno, upaljen depresijski).

6.1. Senzori

Mrežu senzora čine senzori (slika 6-3) ADC0, UPR2, ERC4, LOQ5, SHZ1 i SPT3. Svi senzori osim ADC0 bilježe podatke o temperaturi, relativnoj vlazi i tlaku u određenom vremenu. Senzor ADC0 se od ostalih razlikuje u tome što ne bilježi podatke o tlaku. Razlog tome je što se navedeni senzor nije opremljen za mjerenje tlaka. Lokacija tog senzora je veoma bitna jer nam daje vjerodostojne podatke o atmosferskim uvjetima na samoj lokaciji rudnika. Zbog toga dobivamo točne rezultate, za razliku da smo uzeli podatke od DHMZ za područje grada Samobora.



Slika 6-3 Vanjski senzor Rotronc HF1



Slika 6-4. Unutarnji senzor

Unutarnji senzori ERC4, LOQ5, SHZ1, SPT3, UPR2 izrađeni su na Rudarsko-geološko-naftnom fakultetu.

7. REZULTATI OBRADU U PYTHONU

Nakon što smo prikupili podatke od senzora i prebacili ih na računalo možemo započeti sa pripremom podataka koje želimo analizirati.

7.1. Učitavanje txt.datoteke u Python

Kako bi Python znao na koji način učitava podatke i kako ih sprema, moramo deklarirati stupce i senzore unutar njih. To ćemo učiniti tako što ćemo napraviti strukturu podataka kojom će onda računalo znati raspoznati senzore i ventilator :

```
structure1 = {
    'ADC0': {'T': np.nan, 'RH': np.nan},
    'SHZ1': {'T': np.nan, 'RH': np.nan, 'P': np.nan},
    'UPR2': {'T': np.nan, 'RH': np.nan, 'P': np.nan},
    'SPT3': {'T': np.nan, 'RH': np.nan, 'P': np.nan},
    'ERC4': {'T': np.nan, 'RH': np.nan, 'P': np.nan},
    'LOQ5': {'T': np.nan, 'RH': np.nan, 'P': np.nan},
    'FAN' : {'State': np.nan}
}
```

S ovime smo zapravo napisali da senzor određenog imena predstavlja „ključ“ u kojem spremamo vrijednosti temperature i relativne vlage (T i RH). Kada program prolazi kroz datu on će prepoznati senzor jer smo ga imenovali i pridružiti mu vrijednost temperature i relativne vlage u određenom trenutku. Kod ostalih senzora spremamo i dodatnu vrijednost; tlak, a kod ventilatora njegovo stanje (0,1,-1). tek kad smo napravili strukturu, možemo učitati podatke :

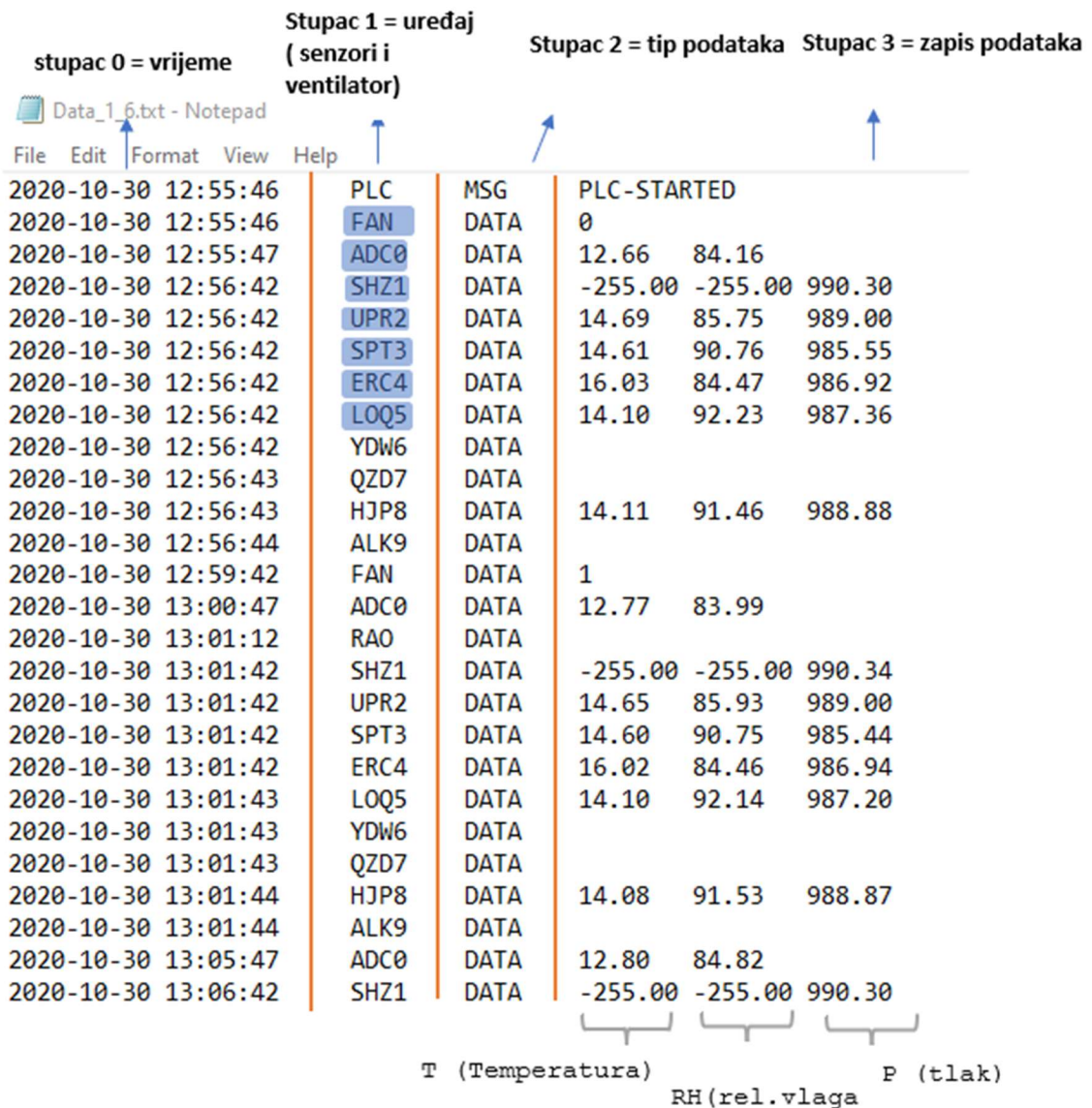
```
input_data1 = {}
file = open('C:/Users/david/OneDrive/Desktop/Diplomski
rad/Data_1_6.txt', 'r')
```

Ovime se učitava polje podataka, odnosno cijeli file data1. Sa file = open dajemo računalu adresu, odnosno mjesto gdje smo spremili datoteku kako bih ga mogao uzeti i učitati. Unutar ulaznih podataka moramo odrediti što su stupci u datoteci (slika 7-1) predstavljaju :

```
time = column[0]
device = column[1]
data_type = column[2]
```

```
data_record = column[3]
```

stupac 0 = vrijeme Stupac 1 = uređaj (senzori i ventilator) Stupac 2 = tip podataka Stupac 3 = zapis podataka



stupac 0 = vrijeme	Stupac 1 = uređaj (senzori i ventilator)	Stupac 2 = tip podataka	Stupac 3 = zapis podataka
2020-10-30 12:55:46	PLC	MSG	PLC-STARTED
2020-10-30 12:55:46	FAN	DATA	0
2020-10-30 12:55:47	ADC0	DATA	12.66 84.16
2020-10-30 12:56:42	SHZ1	DATA	-255.00 -255.00 990.30
2020-10-30 12:56:42	UPR2	DATA	14.69 85.75 989.00
2020-10-30 12:56:42	SPT3	DATA	14.61 90.76 985.55
2020-10-30 12:56:42	ERC4	DATA	16.03 84.47 986.92
2020-10-30 12:56:42	LOQ5	DATA	14.10 92.23 987.36
2020-10-30 12:56:42	YDW6	DATA	
2020-10-30 12:56:43	QZD7	DATA	
2020-10-30 12:56:43	HJP8	DATA	14.11 91.46 988.88
2020-10-30 12:56:44	ALK9	DATA	
2020-10-30 12:59:42	FAN	DATA	1
2020-10-30 13:00:47	ADC0	DATA	12.77 83.99
2020-10-30 13:01:12	RAO	DATA	
2020-10-30 13:01:42	SHZ1	DATA	-255.00 -255.00 990.34
2020-10-30 13:01:42	UPR2	DATA	14.65 85.93 989.00
2020-10-30 13:01:42	SPT3	DATA	14.60 90.75 985.44
2020-10-30 13:01:42	ERC4	DATA	16.02 84.46 986.94
2020-10-30 13:01:43	LOQ5	DATA	14.10 92.14 987.20
2020-10-30 13:01:43	YDW6	DATA	
2020-10-30 13:01:43	QZD7	DATA	
2020-10-30 13:01:44	HJP8	DATA	14.08 91.53 988.87
2020-10-30 13:01:44	ALK9	DATA	
2020-10-30 13:05:47	ADC0	DATA	12.80 84.82
2020-10-30 13:06:42	SHZ1	DATA	-255.00 -255.00 990.30

T (Temperatura) RH (rel. vlaga) P (tlak)

Slika 7-1. Struktura podataka

Dakle python od sad podatke u stupcu 0 gleda kao vrijeme, stupac 1 gleda kao uređaj, stupac 2 kao tip podataka te stupac 3 kao zapis podataka, odnosno njihova vrijednost. Kad nam recimo bude potrebno vrijeme za prikazivanje dijagrama, pozvati ćemo `time` i on će znati da je to stupac koji prikazuje vrijeme i povući će te podatke. Međutim prije samog učitavanja moramo pozvati modul koji koristi python za rad s nizovima i matricama kao što je i ova naša datoteka. Modul se zove Numpy, a poziva se :

```
import numpy as np
```

Za data2 koristimo isti postupak. Cijeli kod za učitavanje podataka može se vidjeti u prilogu 3.

7.2. Kretanje temperature, relativne vlage i rosišta kroz godinu

Za svaki senzor smo odredili uz pomoć temperature i relativne vlage, po formuli 3-3 temperaturu rosišta kroz cijelu godinu (prilog 2). Za dobivanje podataka o temperaturi i relativnoj vlazi kroz učitani data1 i data 2 smo postavili for petlju koja prolazi kroz cijelu datoteku te unutar nje if uvjet koji provjerava je li istina da je podatak o određenom senzoru u toj datoteci u tom vremenu; ako je zapisuje ga i sprema u novu varijablu :

```
for a,b in input_data1.items():  
  
    if 'FAN' in b and 'State' in b['FAN']:  
        Fan_1 = b['FAN']['State']  
    if 'ADC0' in b and 'T' in b['ADC0']:  
        Temperatura_ADC0 = b['ADC0']['T']  
    if 'ADC0' in b and 'RH' in b['ADC0']:  
        Rel_Vlaga_ADC0 = b['ADC0']['RH']
```

Važno je da if uvjet bude unutar for petlje kao što je gore i prikazano. Za sve ostale senzore postupak je isti. Dakle, podatak o temperaturi i vlazi spremljen je u varijable, no prije toga iznad for petlje moramo formirati polje s listama u koje će se te varijable spremati. Znači gleda se interval od 120 sekundi, u svakom intervalu imamo novi podatak o temperaturi i vlazi te se on sprema u to polje s listama :

```
Fan = []  
ADC0 = {'T': [], 'RH': [], 'DP': []}
```

Ventilator ima samo jednu listu jer sprema samo stanja 0,1 i -1, dok senzori imaju više lista, sa podacima o temperaturi, vlazi, tlaku itd. Prikazivanje grafa se može napraviti na više načina, ovdje je odabran PyQtGraph. To je grafika koja je napravljena na Numpy i koristi se za prikazivanje znanstvenih, matematičkih i inženjerskih podataka. Prvo ga treba instalirati na računalo (u windows shellu) naredbom :


```
pip install pyqtgraph
```

Kad je instaliran, još ga treba pozvati u programskom kodu u kojem pišemo naredbom :

```
import pyqtgraph as pg
```

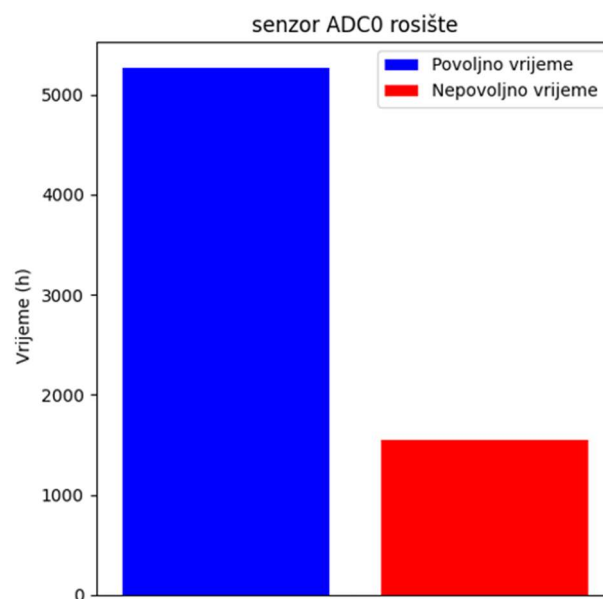
Konačno, za prikazivanje grafa odredimo koje podatke želimo prikazivati. Apscisa (x-os) uvijek prikazuje vrijeme, a ordinata (y-os) se mijenja s obzirom što želimo prikazati (temperatura, vlaga, rosište). U prilogu 2, na grafu temperature možemo primijetiti nekoliko stvari. Vanjski senzor ADC0 uvijek pokazuje veće ili manje temperature od ostalih senzora koji su unutar rudnika, zavisno o godišnjem dobu. Tako recimo od listopada pa sve do ožujka, temperature vanjskog senzora se kreću između 0 i 10° C, a u 5. mjesecu dolazi do skoka temperature koji traje sve do 9. mjeseca, gdje je vanjska temperatura uvijek iznad 20°C. To je i očekivano jer je ljeto. Temperatura u rudniku je približno konstantna kao što je najbolje vidljivo u periodu između lipnja i rujna gdje se svi unutarnji senzori kreću oko 12,5 °C. Jedino zamjetno odstupanje pokazuje senzor LOQ5 koji u zimskim mjesecima pokazuje manje temperature (5 do 8°C). Mogući razlog tome je što se senzor nalazi blizu izlaza iz rudnika. U prilogu 2 na grafu relativne vlažnosti možemo primjetiti kako vanjski senzor ADC0 pokazuje velika odstupanja od unutarnjih senzora, pogotovo u ljetnim mjesecima. Naime vlaga na senzoru ADC0 je znatno manja nego kod unutarnjih senzora. To je i očekivano jer rudnici su ipak mjesta veće vlage. Možemo primijetiti kako unutarnja vlaga kod svih senzora je približno ista (od 80% pa nadalje). Jedino odstupanje prikazuje senzor SHZ1 u veljači i ožujku gdje je nešto manje nego ostali unutarnji senzori. Razlog tome može biti mjesto senzora, koji se nalazi blizu ulaza u rudnik. Na grafu rosišta primjećujemo da najmanju temperaturu rosišta preko zimskih mjeseci prikazuje vanjski senzor ADC0, a preko ljetnih najveću. Unutarnji senzori preko ljeta imaju približne temperature rosišta, odstupanja možemo vidjeti u listopadu, prosincu, veljači i ožujku.

7.2.1. Povoljno i nepovoljno rosište

Vanjska temperatura, odnosno temperatura rosišta utječe na vlagu u rudniku. Prosječna temperatura u rudniku iznosi 12,5 °C. Dakle ako je vanjsko rosište manje od 12,5 °C, to je povoljno jer do kondenzacije dolazi na manjoj temperaturi nego što je ona u rudniku. Nepovoljna situacija dolazi kad je vanjska temperatura rosišta jednaka ili iznad 12,5°C. S obzirom da imamo podatke o vanjskoj temperaturi rosišta zbog senzora ADC0 koji očitava temperaturu zraka i relativnu vlažnost zraka; možemo vidjeti kad imamo povoljnu, a kad nepovoljnu situaciju. U programu trebamo samo provjeriti kad je veća, a kad manja od 12,5°C :

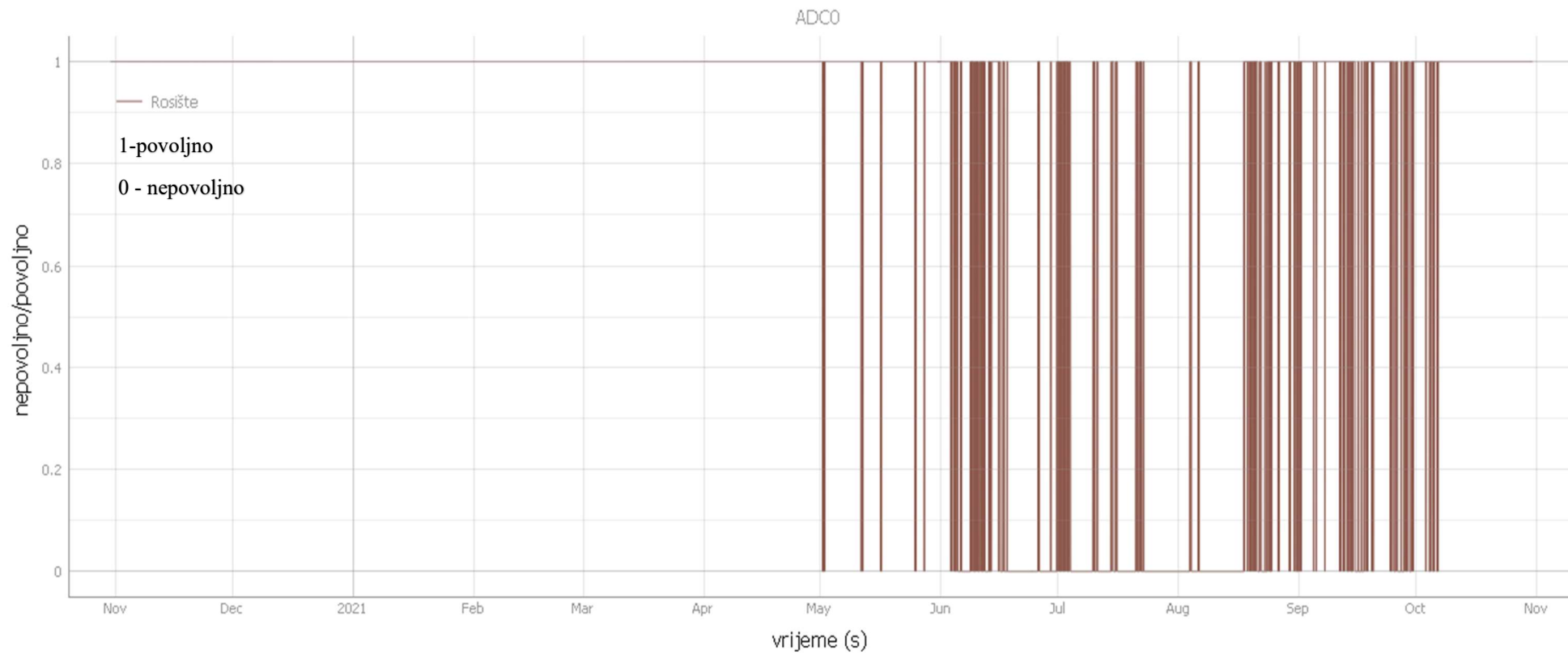
```
if DP < 12.5:
    Uvjeti_1 = 1
if DP >= 12.5:
    Uvjeti_1 = 0
if Uvjeti_1 == 1:
    Povoljno_ADC0 = Povoljno_ADC0 + interval
if Uvjeti_1 == 0:
    Nepovoljno_ADC0 = Nepovoljno_ADC0 + interval
```

DP se odnosi na temperaturu rosišta. Svaki put kad je veća od 12,5 sprema se vrijednost i dodaje interval i time dobijemo ukupno vrijeme kad je povoljno, odnosno nepovoljno kao što se može vidjeti na slici 7-2.



Slika 7-2. Povoljno i nepovoljno vrijeme

Ukupno povoljno vrijeme iznosi 5263.5 sati, a nepovoljno 1547.6 sati. S ovime smo dokazali potrebu ventilatora u rudniku Sv. Barbare jer ima nepovoljnih vremena. Da nema sustava ventilacije došlo bi do kondenzacije vlage unutar rudnika u tom periodu kad je nepovoljno rosište (taj period iznosi 64 dana). Nadalje, možemo prikazati u kojim mjesecima je bilo najviše nepovoljnih intervala ili najviše povoljnih (slika 7-3.).na slici možemo primijetiti kako kroz većinu zimskog perioda imamo konstanto povoljnu temperaturu rosišta (od listopada do ožujka). Nepovoljni intervali javljaju se u ljetnim mjesecima gdje su i temperature zraka više.



Slika 7-3. Povoljno/ nepovoljno kroz mjesece

Sljedeću stvar koju možemo vidjeti jest temperaturu rosišta na unutarnjim sensorima. To će nam pokazati je li ikada došlo do neželjene pojave, a to je kondenzacije vlage unutar rudnika. Do kondenzacije bi došlo ako je temperatura rosišta na senzoru jednaka ili veća od temperature na senzoru u određenom trenutku. (drugim riječima, ako je relativna vlaga bila 100 %). Dok je temperatura rosišta na senzoru manja od temperature rosišta neće doći do kondenzacije vlage. To provjeravamo na isti način kao i vanjski senzor, samo što ovdje stavljamo temperaturu senzora, a ne 12,5 °C :

```

if DP_1 < Temperatura_UPR2:
    Uvjeti_2 = 1
if DP_1 >= Temperatura_UPR2:
    Uvjeti_2 = 0
if Uvjeti_2 == 1:
    Povoljno_UPR2_2 = Povoljno_UPR2_2 + interval
if Uvjeti_2 == 0:
    Nepovoljno_UPR2_2 = Nepovoljno_UPR2_2 + interval

```

DP1 i ovdje predstavlja temperaturu rosišta senzora, u ovom slučaju senzora UPR2. Za sve druge senzore postupak je isti. Dobivamo sljedeće rezultate (slika 7-4) :

```

    Za cijelu godinu (od 11.mj 2020 do 11. mj 2021) (data 1 + data2) :
Povoljno UPR2 (h) 6811.133333333333
Povoljno SHZ1 (h) 6811.133333333333
Povoljno ERC4 (h) 6811.133333333333
Povoljno LOQ5 (h) 6811.133333333333
Povoljno SPT3 (h) 3796.0
.....
Nepovoljno UPR2 (h) 0
Nepovoljno SHZ1 (h) 0
Nepovoljno ERC4 (h) 0
Nepovoljno HJP8 (h) 0
Nepovoljno LOQ5 (h) 0
Nepovoljno SPT3 (h) 0

```

Slika 7-4. povoljno/nepovoljno unutarnji senzori

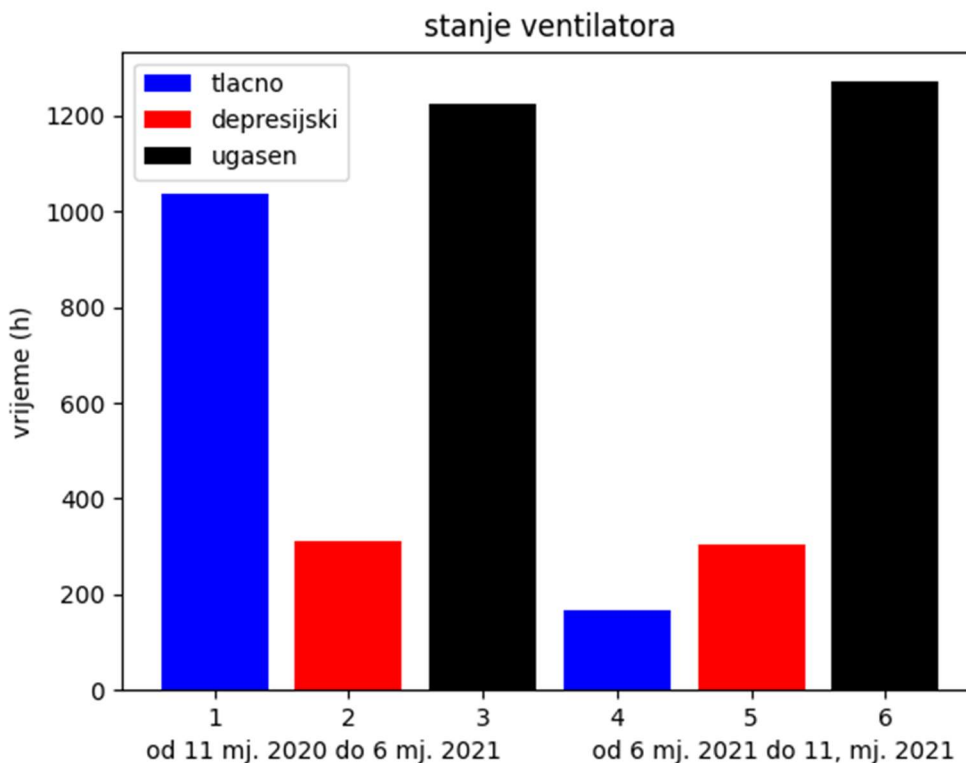
Ovime smo pokazali da nikada nije došlo do kondenzacije unutar rudnika, uvijek je temperatura na senzoru bila manje od temperature rosišta. Ventilacija je uzrok tome što nikada nije došlo do kondenzacije.

7.3. Rad ventilatora

Kao što smo rekli, ventilator ima mogućnost rada na više načina, tlačno i depresijski (poglavlje 4). Koristeći programski kod možemo prikazati koliko vremena je bio ugašen, koliko radio tlačno, odnosno depresijski. Treba nam uvjet if unutar for petlje koji će provjeravati uvjete :

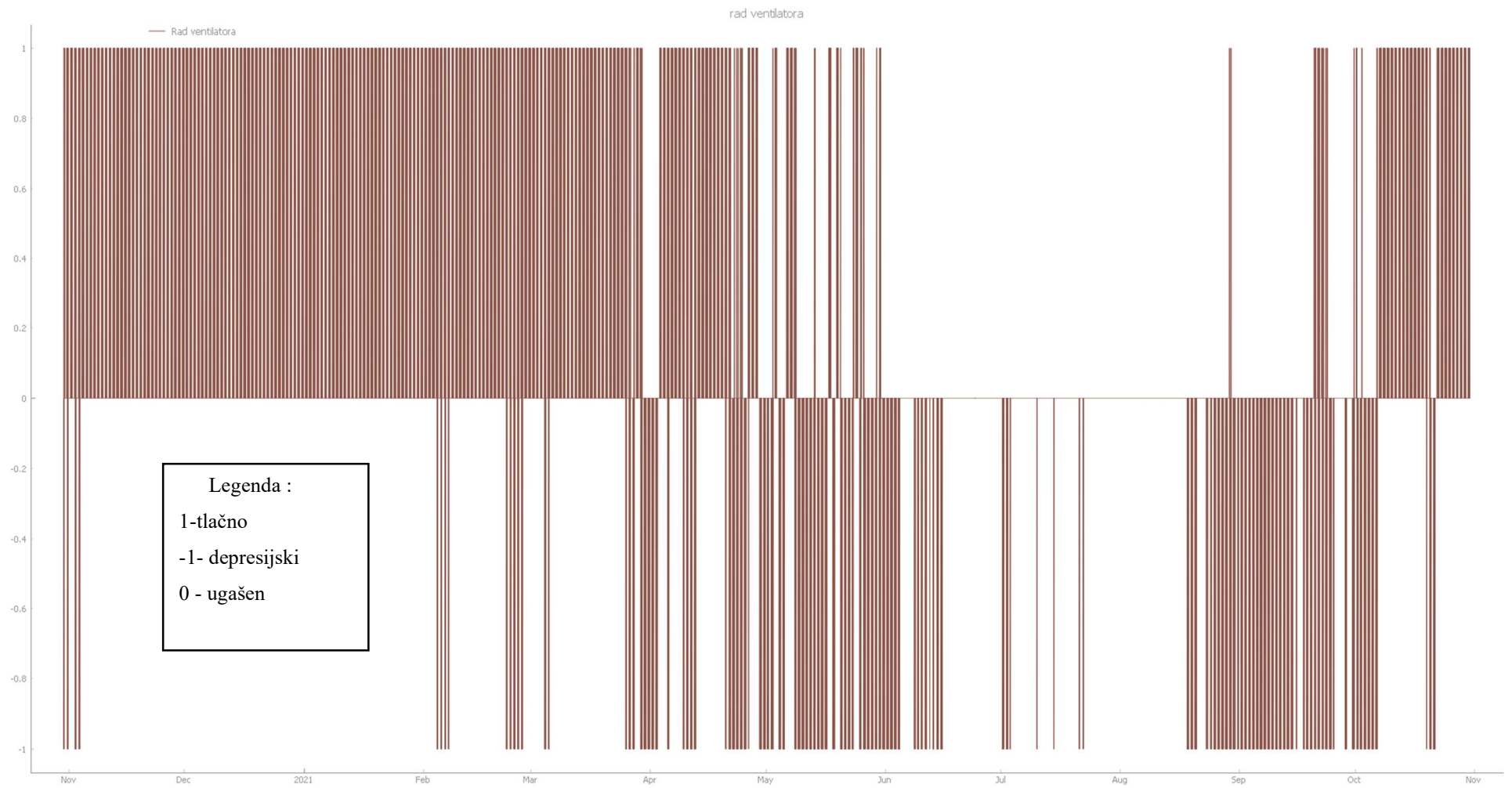
```
if Fan_1 == 1 :  
    Tlacno = Tlacno + (interval/3600)  
  
if Fan_1 == 0 :  
    Ugasen = Ugasen + (interval/3600)  
  
if Fan_1 == -1 :  
    Depresija = Depresija + (interval/3600)
```

Za tlačni način provjerava kad je status ventilatora 1, za depresijski -1, a za ugašen 0. Svaki put kad nađe taj podatak sprema ga u varijablu i dodaje interval. Time smo postigli prikazati ukupno vrijeme različitog načina rada ventilatora (slika 7-5).



Slika 7-5. Rad ventilatora

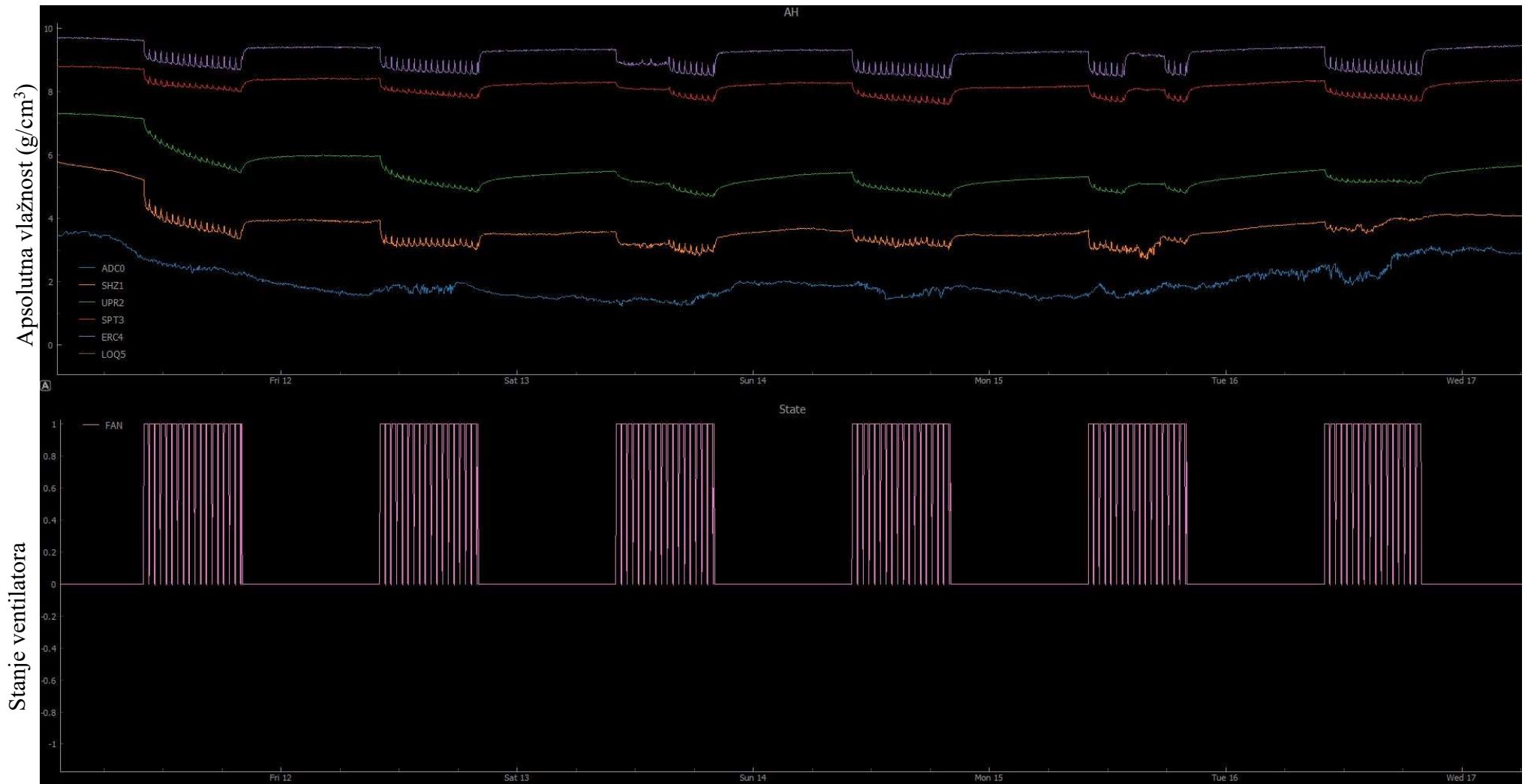
Vidimo da je ventilator radio više tlačnim načinom nego depresijskim, i to u zimskim mjesecima. U ljetnim mjesecima ventilator radi više depresijski nego tlačno. To je zbog promjene smjera strujanja toplinske depresije ljeti u odnosu na zimu. Napomena : slika 7-5 prikazuje ukupno vrijeme ali preko dana, ventilator je uvijek bio isključen preko noći i to vrijeme nismo uzeli u obzir. Bolji prikaz rada ventilatora po određenim mjesecima u godini prikazuje slika 7-6. Ventilator je između studenog i ožujka radio tlačno i bio ugašen, gotovo pa su se konstanto izmjenjivali. Ventilator bi radio oko pola sata te bi onda bio ugašen 20-ak minuta, a preko noći bi bio ugašen (oko 8h u komadu). Oko travnja počinje raditi i depresijskim načinom, a zanimljivo je da gotovo pa cijelo vrijeme između lipnja i kolovoza ventilator ne radi tj. ugašen je.



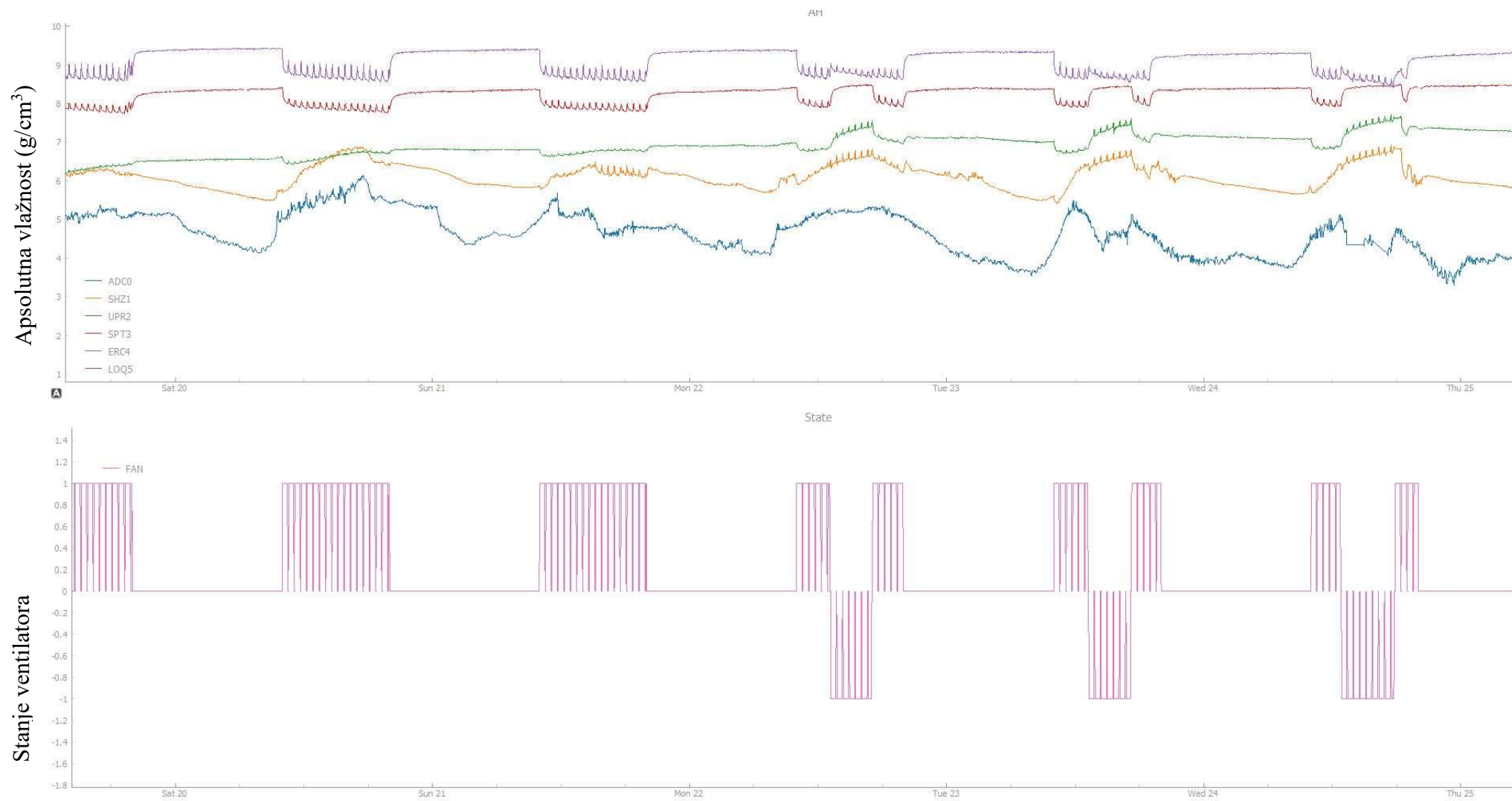
Slika 7-5. Rad ventilatora kroz vrijeme

7.3.1. Utjecaj Ventilacije na apsolutnu vlagu unutar rudnika

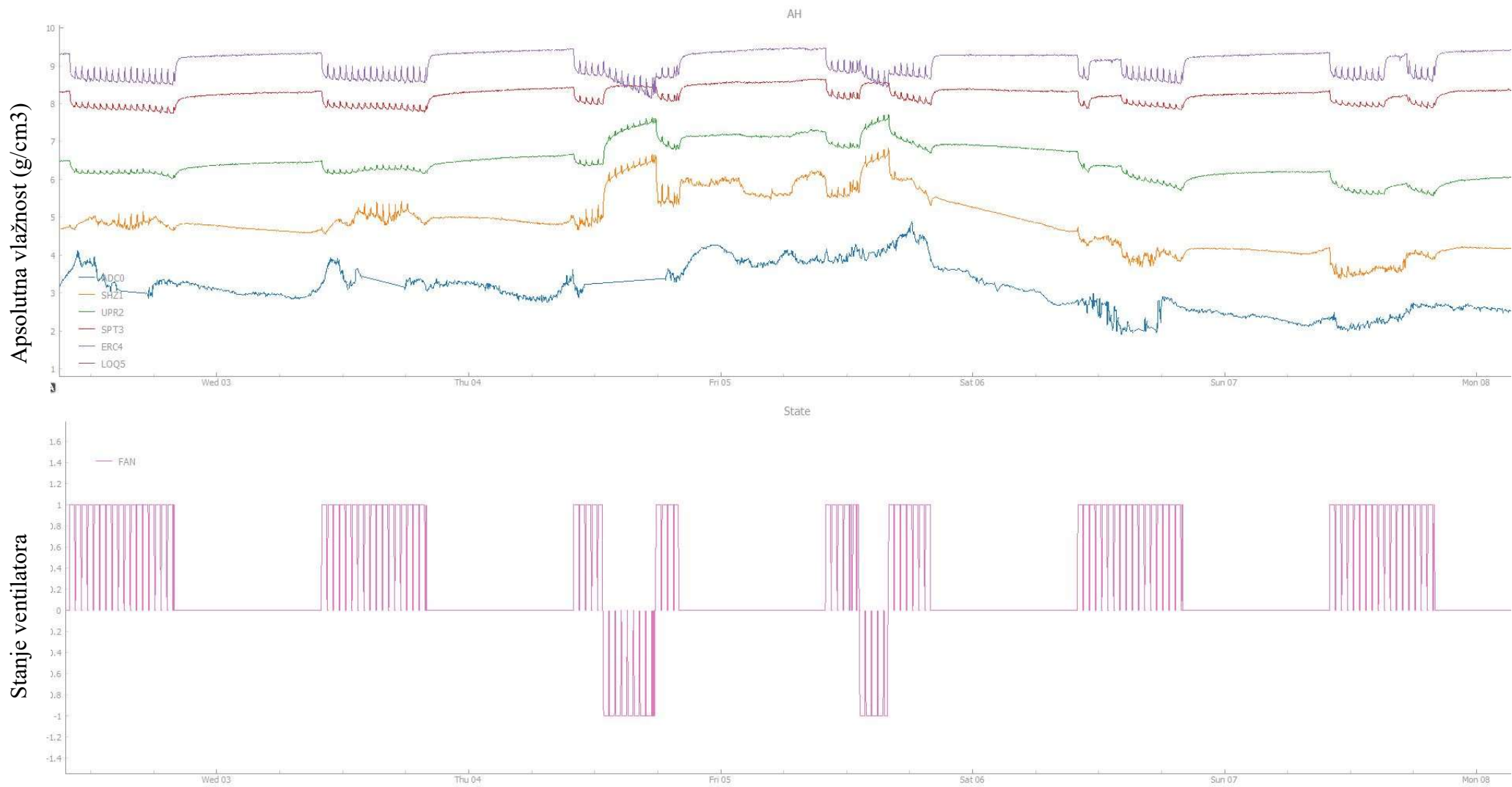
Nakon što smo odredili načine rada ventilatora kroz vrijeme, možemo prikazati radi li ventilacija kako bi trebala. Dakle korištenjem ventilacije očekujemo smanjenje apsolutne vlažnosti unutar rudnika. To možemo vidjeti na slici 7-6. Zbog gustih podataka, nije moguće prikazati za cijelu godinu, stoga smo uzeli jedan vremenski interval (6 dana). Možemo primijetiti kako apsolutna vlaga opada kad se ventilator upali. Senzor ADC0 se ne mijenja uključivanjem jer se on nalazi izvan rudnika. Ostali senzori (unutarnji senzori) jasno pokazuju da se vlažnost smanjuje kad se ventilator upali. U ovom promatranom intervalu možemo vidjeti da ventilator radi tlačno i ugašen je. To je bitna stavka jer nam objašnjava kako je na nekim sensorima vlaga veća nego na drugima. Tako je recimo na senzoru SHZ1 vlaga najmanja (od 4 do 6 g/cm³), dok je najviša na sensorima ERC4 i LOQ5 (oko 10 gm/cm³). To je zbog toga što se senzor SHZ1 nalazi najbliže ulazu svježeg zračnog struje (ventilator radi tlačno i upuhuje zrak u rudnik, a senzor ADC0 se nalazi najbliže ulazu u rudnik) te zrak tada prolazi kroz rudnik i vuče vlagu kroz rudnik dok ne izađe van. Zbog toga senzori koji se nalaze bliže izlazu bilježe nešto veće količine vlage. Ovo su idealni uvjeti u kojima vlaga dok je ventilator uključen pada na svim sensorima. U različitim periodima godine postoje i slučajevi gdje apsolutna vlažnost pada na sensorima bliže ulazu zračne struje a poraste na sensorima bliže izlazu zračne struje (slika 7-7). Slika pokazuje dijagram na kojem se može vidjeti da vlaga ne opada na svim sensorima kad je ventilator upaljen. Zbog porasti vanjske vlage (senzor ADC0) dolazi do porasti vlage na senzoru SHZ1, neovisno što ventilator radi. Uzrok tome je što se senzor SHZ1 nalazi blizu ulaza u rudnik pa je zbog vanjskog skoka vlage i senzor SHZ1 podložan tom trendu. Takav, ali blaži slučaj pokazuje i senzor UPR2 (nalazi iza SHZ1) Na ostale senzore takva pojava nema utjecaj. Slika 7-8. prikazuje da način rada ventilatora utječe na količinu vlage na pojedine senzore. Tako da senzor SHZ1 kad ventilator radi tlačno ima manju vlagu nego kad ventilator radi depresijski. U depresijskom načinu rada senzor bilježi nagli skok vlage (sa 4 na 6,5 gm/cm³). razlog tome je što se promijenio smjer strujanja zraka pa je sad senzor SHZ1 zadnji po redu, odnosno dolazi mu zrak koji je već prošao kroz rudnik i pokupio vlagu u njemu. Zbog toga senzori koji su bliže izlazu iz rudnika, u trenutku kad ventilator radi depresijski, oni bilježe manje iznose vlage nego kad ventilator radi tlačno. To se vidi na senzoru ERC4 (otprilike 1-2 g/cm³ razlike). Prosječno smanjenje vlage po dnevnoj bazi (razlika dan-noć) iznosi otprilike 2 g/cm³ ovisno koji senzor promatramo



Slika 7-6. Utjecaj ventilacije na apsolutnu vlagu unutar rudnika (idealni slučaj)



Slika 7-7. Utjecaj ventilacije na apsolutnu vlažnost unutar rudnika (karakteristični intervali 1)



Slika 7-8. Utjecaj ventilacije na apsolutnu vlažnost unutar rudnika (karakteristični intervali)

8. ZAKLJUČAK

Korištenjem programskog jezika Python uspješno smo prikazali kretanje temperature zraka, relativne vlage, apsolutne vlage i temperature rosišta kroz godinu dana u rudniku Sv. Barbare. Zapazili smo kako je relativna vlaga unutar rudnika uvijek iznad 50% (kreće se između 75 do preko 90%), prosječna temperatura u rudniku je oko 12,5 °C. Postavljeni senzori unutar rudnika (ERC4, LOQ5, SHZ1, SPT3,UPR4) te vanjski senzor (ADC0) omogućili su nam da detaljnije opišemo stanja u rudniku. Tako smo korištenjem algoritama pokazali na temelju očitavanja vanjskog senzora da postoje nepovoljni vremenski intervali (ukupno 64 dana u godini) u kojoj može doći do nepoželjne pojave; kondenzacije vodene pare unutar rudnika. Time se opravdali potrebu sustava ventilacije unutar rudnika. Da ventilacije nema, došlo bi do kondenzacije i time bi se degradirala drvena podgrada unutar rudnika. Na temelju podataka zabilježenih unutarnjim sensorima dokazali smo da do kondenzacije unutar rudnika nije došlo (tj. relativna vlaga nikad nije bila 100%). To znači da je sustav ventilacije radio svoju „zadaću“, odvlaživao je rudnik svojim radom. Također prikazali smo kako je ventilator radio kroz vrijeme, tlačno, depresijski i kada je bio ugašen. Više vremena radio je tlačno, pogotovo u zimskim mjesecima, dok je u ljetnim više radio depresijskim načinom. To je zbog toga što se je smjer strujanja zraka kroz rudnik mijenjao zbog toplinske depresije koja je uzrokovala tu promjenu pri različitim vremenskim dobima. Zatim smo pokazali kako se apsolutna vlaga unutar rudnika mijenja radom ventilatora. Kada je ventilator upaljen, apsolutna vlaga pada, a kada je ugašen ona raste (takav je slučaj u idealnim uvjetima, zbog gustoće podataka, nemoguće je prikazati kretanju kroz cijelu godinu). Kako bi pokazali da nije uvijek riječ o idealnim uvjetima gdje vlaga pada na svim sensorima kad ventilator radi, prikazali smo karakteristične intervale gdje vlažnost pada na sensorima bliže ulazu zračne struje, a poraste na sensorima bliže izlazu zračne struje, zavisno kojim načinom ventilator radi (depresijski ili tlačno). Prosječno smanjenje apsolutne vlage na dnevnoj bazi je iznosilo oko 2 g/cm³. Dakle ventilator je opravdan u rudniku te je dobro optimiziran. Također treba spomenuti kako bez korištenja programskog jezika ovakva analiza ne bi bila moguća, te kao inženjeri rudarstva stalno trebamo pratiti nove trendove u znanosti i tehnologiji te ih pokušati implementirati u naš rad kako bi ostvarili maksimalnu učinkovitost i točnost.

9. LITERATURA

HARTMAN, H., MUTMANSKY, J., RAMANI, R., WANG, Y. 1997. Mine Ventilation and air conditioning. Third edition. Toronto. John Wiley & Sons.

HRUŠKA, M. 2018. Osnove programiranja (Python.) Sveučilište u Zagrebu. Sveučilišni računski centar. Zagreb.

Hrvatska enciklopedija. 2020. Rosište . Mrežno izdanje. Leksikografski zavod Miroslav Krleža. URL : <https://www.enciklopedija.hr/natuknica.aspx?ID=53391> (25. 10. 2022).

KIUSALAAS, J. 2010. Numerical Methods in Engineering with Python. New York. Cambridge University Press.

KLANFAR, M. 2018. Vjetrenje i odvodnjavanje. Predavanja. Sveučilište u Zagrebu: Rudarsko-geološko-naftni fakultet. Zagreb

LAWRENCE, M. 2005. The relationship between relative humidity and the dewpoint temperature in moist air: A simple conversion and applications. Bulletin of the American Meteorological Society. 86(2). str. 225-233.

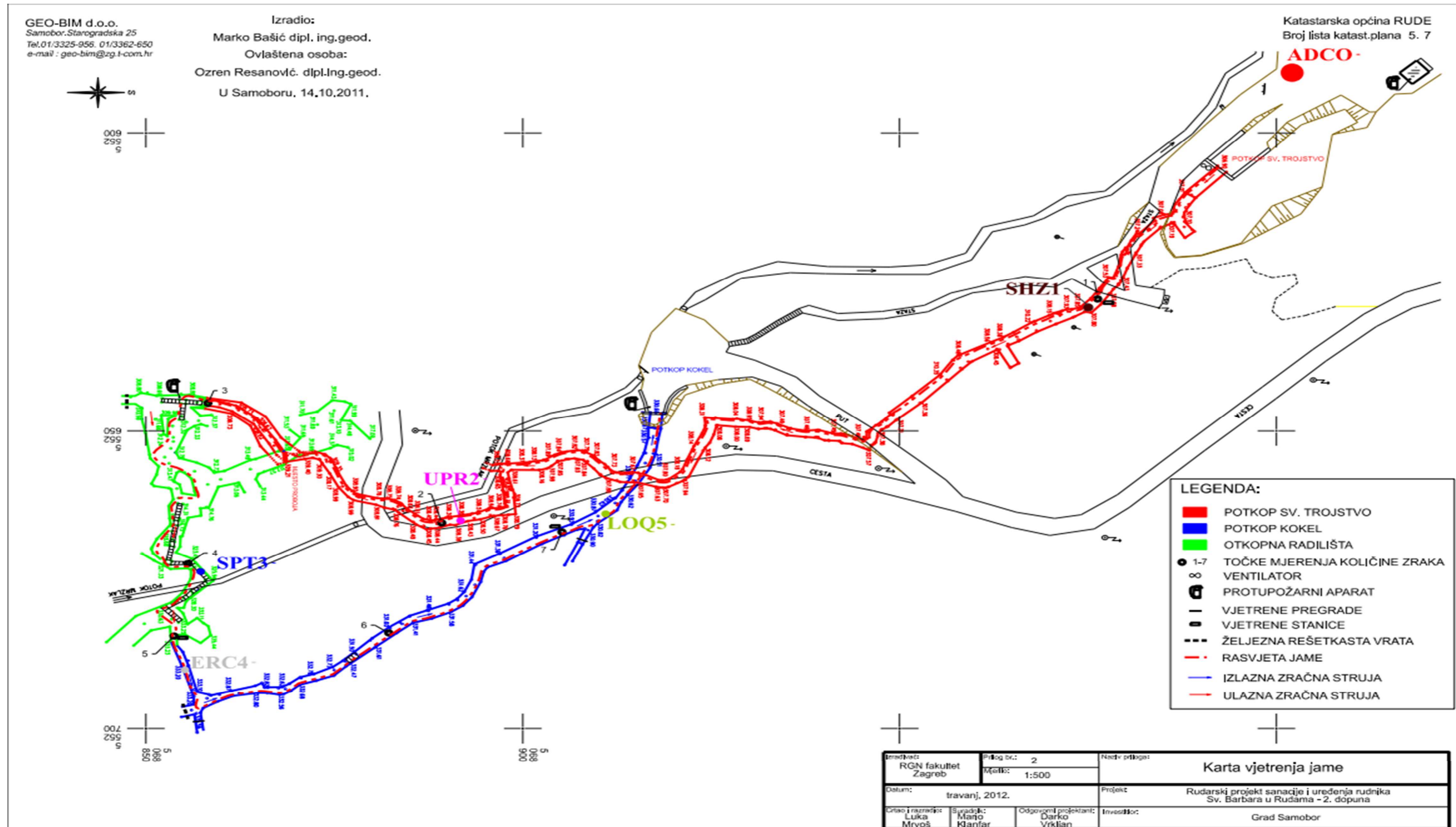
MEDVED, S. 2021. Building Physics : Heat, Ventilation, Moisture, Light, Sound, Fire, and Urban Microclimate. Ljubljana. Springer Tracts in Civil Engineering.

VRKLJAN, D. 2007. Ventilacija vježbe-primjeri zadataka. Sveučilište u Zagrebu. Rudarsko-geološko-naftni fakultet. Zagreb

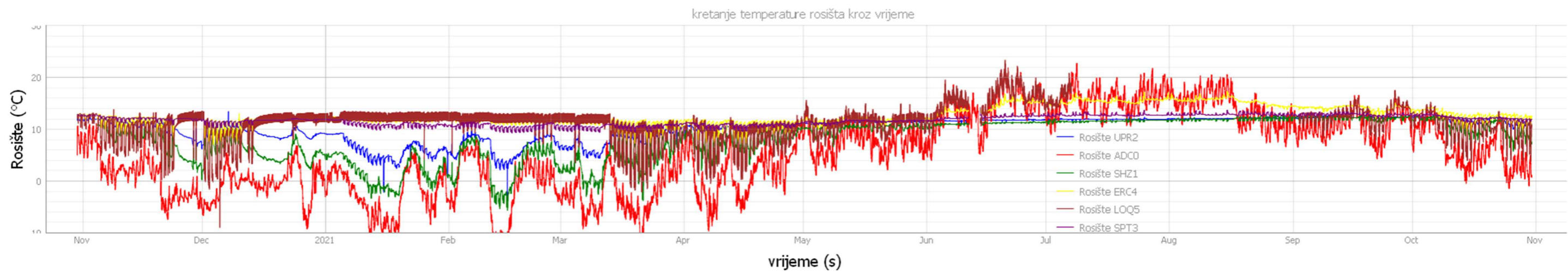
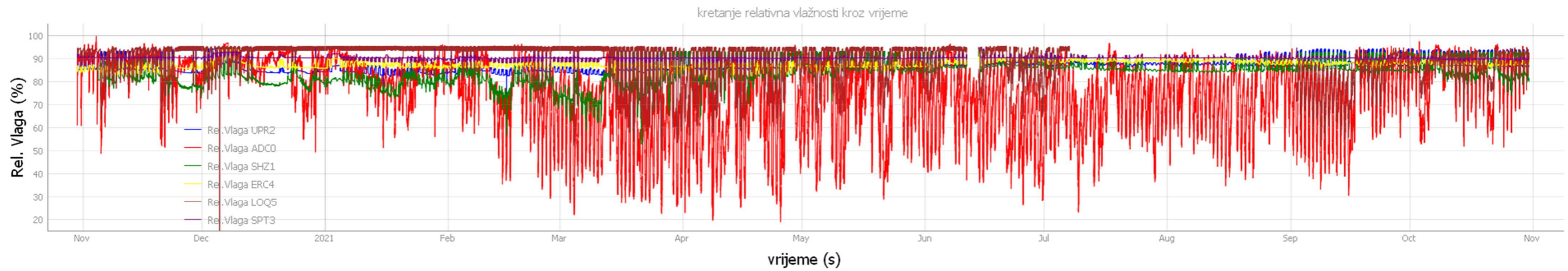
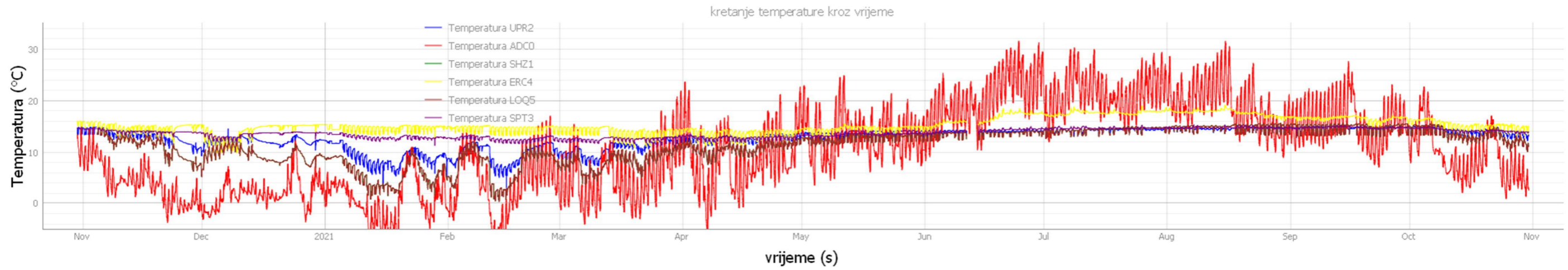
KUD Oštrc. 2019. Rudnik sveta Barbara. URL:

<https://www.rudnik.hr/hr/news/nadogradnja-sustava-vjetrenja/> (18.9.2022)

Prilog 1. Karta vjetrenja sa lokacijama senzora



Prilog 2. Temperatura, rosište i relativna vlaga



Prilog 3. Programski kod u Pythonu

```
import datetime
import numpy as np
import pyqtgraph as pg

structure1 = {
    'ADC0':{'T':np.nan, 'RH':np.nan},
    'SHZ1':{'T':np.nan, 'RH':np.nan, 'P':np.nan},
    'UPR2':{'T':np.nan, 'RH':np.nan, 'P':np.nan},
    'SPT3':{'T':np.nan, 'RH':np.nan, 'P':np.nan},
    'ERC4':{'T':np.nan, 'RH':np.nan, 'P':np.nan},
    'LOQ5':{'T':np.nan, 'RH':np.nan, 'P':np.nan},
    'HJP8':{'T':np.nan, 'RH':np.nan, 'P':np.nan},
    'FAN' :{'State': np.nan}
}

input_data1 = {}

file = open('C:/Users/david/OneDrive/Desktop/Diplomski rad/Data_1_6.txt',
'r')
data = []
lines = file.readlines()
for line in lines:
    data.append(line.split('\t'))
for column in data:
    if len(column[0]) == 19 and len(column) >= 3: #Check if timestamp is
present and line has at least time, device and data_type column,
otherwise it is an error
        #time = datetime.datetime.strptime(column[0], "%Y-%m-%d
%H:%M:%S").timestamp()
        time = column[0]
        device = column[1]
        data_type = column[2]
        data_record = column[3:]
        if device in structure1:
            if time not in input_data1:
                input_data1.update({time:{}})
            if device not in input_data1[time]:
                input_data1[time].update({device:structure1[device].copy()})
```

```

        if len(data_record) == len(structure1[device]):
            for count, record in enumerate(data_record):
                try:
                    record1 = float(record)

input_data1[time][device][list(input_data1[time][device].keys())[count]]
= record1

                except:

input_data1[time][device][list(input_data1[time][device].keys())[count]]
= np.nan

                    input_data1[time][device].update({'Err':'value'})

        if len(data_record) != len(structure1[device]):
            input_data1[time][device].update({'Err':'record'})

    else:
        input_data1[list(input_data1.keys())[-
1]].update({'Err':{'Err':'line'}})

file.close()
print('file loaded')

input_data2 = {}

file = open('C:/Users/david/OneDrive/Desktop/Diplomski rad/Data_2_6.txt',
'r')
data = []
lines = file.readlines()
for line in lines:
    data.append(line.split('\t'))
for column in data:
    if len(column[0]) == 19 and len(column) >= 3:    #Check if timedata is
present and line has at least time, device and data_type column,
otherwise it is an error
        #time = datetime.datetime.strptime(column[0], "%Y-%m-%d
%H:%M:%S").timestamp()
        time = column[0]

```

```

device = column[1]
data_type = column[2]
data_record = column[3:]
if device in structure1:
    if time not in input_data2:
        input_data2.update({time:{}})
    if device not in input_data2[time]:

input_data2[time].update({device:structure1[device].copy()})
    if len(data_record) == len(structure1[device]):
        for count, record in enumerate(data_record):
            try:
                record1 = float(record)

input_data2[time][device][list(input_data2[time][device].keys())[count]]
= record1

            except:

input_data2[time][device][list(input_data2[time][device].keys())[count]]
= np.nan

                input_data2[time][device].update({'Err':'value'})

        if len(data_record) != len(structure1[device]):
            input_data2[time][device].update({'Err':'record'})

    else:
        input_data2[list(input_data2.keys())[-
1]].update({'Err':{'Err':'line'}})

file.close()
print('file 2 loaded')

time = []
interval = 120
Fan = []
ADC0 = {'T':[], 'RH':[], 'DP':[]}
UPR2 = {'T':[], 'RH':[], 'P':[], 'DP_1':[]}
SPT3 = {'T':[], 'RH':[], 'P':[], 'DP_2':[]}
ERC4 = {'T':[], 'RH':[], 'P':[], 'DP_3':[]}

```

```

SHZ1 = {'T':[], 'RH':[], 'P':[], 'DP_4':[]}
LOQ5 = {'T':[], 'RH':[], 'P':[], 'DP_5':[]}
HJP8 = {'T':[], 'RH':[], 'P':[], 'DP_6':[]}
SVI = {'DP_AV':[]}
Fan_1 = 0
Temperatura_UPR2, Temperatura_SPT3, Temperatura_ERC4, Temperatura_ADC0,
Temperatura_SHZ1, Temperatura_LOQ5, Temperatura_HJP8 = 0, 0, 0, 0, 0, 0,
0
Rel_Vlaga_UPR2, Rel_Vlaga_SPT3, Rel_Vlaga_ERC4, Rel_Vlaga_ADC0,
Rel_Vlaga_SHZ1, Rel_Vlaga_LOQ5, Rel_Vlaga_HJP8, = 0, 0, 0, 0, 0, 0, 0
Tlak_UPR2, Tlak_SPT3, Tlak_ERC4, Tlak_SHZ1, Tlak_LOQ5, Tlak_HJP8 = 0, 0,
0, 0, 0, 0
Uvjeti_1, Uvjeti_3, Uvjeti_4, Uvjeti_5, Uvjeti_6, Uvjeti_7 = 0, 0, 0, 0,
0, 0
Uvjeti_2 = 0
DP, DP_1, DP_2, DP_3, DP_4, DP_5, DP_6 = 0, 0, 0, 0, 0, 0, 0
Uvjeti_ADC0, Uvjeti_UPR2, Uvjeti_ERC4, Uvjeti_SHZ1, Uvjeti_SPT3, Uvjeti_LOQ5, U
vjeti_HJP8 = [], [], [], [], [], [], []
DP_AV = 0
P_DP_AV = 0
N_DP_AV = 0
Povoljno_UPR2, Povoljno_ADC0, Povoljno_SPT3, Povoljno_ERC4,
Povoljno_SHZ1, Povoljno_LOQ5, Povoljno_HJP8, = 0, 0, 0, 0, 0, 0, 0
Nepovoljno_UPR2, Nepovoljno_ADC0, Nepovoljno_SPT3, Nepovoljno_ERC4,
Nepovoljno_SHZ1, Nepovoljno_LOQ5, Nepovoljno_HJP8 = 0, 0, 0, 0, 0, 0, 0
Povoljno_UPR2_2, Povoljno_ADC0_2, Povoljno_SPT3_2, Povoljno_ERC4_2,
Povoljno_SHZ1_2, Povoljno_LOQ5_2, Povoljno_HJP8_2, = 0, 0, 0, 0, 0, 0, 0
Nepovoljno_UPR2_2, Nepovoljno_ADC0_2, Nepovoljno_SPT3_2,
Nepovoljno_ERC4_2, Nepovoljno_SHZ1_2, Nepovoljno_LOQ5_2,
Nepovoljno_HJP8_2 = 0, 0, 0, 0, 0, 0, 0
Povoljno_u_satima_UPR2_2, Povoljno_u_satima_ADC0_2,
Povoljno_u_satima_SPT3_2, Povoljno_u_satima_ERC4_2,
Povoljno_u_satima_SHZ1_2, Povoljno_u_satima_LOQ5_2,
Povoljno_u_satima_HJP8_2 = 0, 0, 0, 0, 0, 0, 0
Povoljno_u_satima_UPR2, Povoljno_u_satima_ADC0, Povoljno_u_satima_SPT3,
Povoljno_u_satima_ERC4, Povoljno_u_satima_SHZ1, Povoljno_u_satima_LOQ5,
Povoljno_u_satima_HJP8 = 0, 0, 0, 0, 0, 0, 0
Nepovoljno_u_satima_UPR2, Nepovoljno_u_satima_ADC0,
Nepovoljno_u_satima_SPT3, Nepovoljno_u_satima_ERC4,
Nepovoljno_u_satima_SHZ1, Nepovoljno_u_satima_LOQ5,
Nepovoljno_u_satima_HJP8 = 0, 0, 0, 0, 0, 0, 0

```

```

Nepovoljno_u_satima_UPR2_2, Nepovoljno_u_satima_ADC0_2,
Nepovoljno_u_satima_SPT3_2, Nepovoljno_u_satima_ERC4_2,
Nepovoljno_u_satima_SHZ1_2, Nepovoljno_u_satima_LOQ5_2,
Nepovoljno_u_satima_HJP8_2 = 0, 0, 0, 0, 0, 0, 0
ADC0_PG, UPR2_PG, SHZ1_PG, ERC4_PG, HJP8_PG, LOQ5_PG, SPT3_PG = 0, 0, 0,
0, 0, 0, 0
ADC0_NG, UPR2_NG, SHZ1_NG, ERC4_NG, HJP8_NG, LOQ5_NG, SPT3_NG = 0, 0, 0,
0, 0, 0, 0
Tlacno, Ugasen, Depresija = 0, 0, 0
Tlacno_2, Ugasen_2, Depresija_2 = 0, 0, 0
for a,b in input_data1.items():

```

```

    if 'FAN' in b and 'State' in b['FAN']:
        Fan_1 = b['FAN']['State']

```

```

    if 'ADC0' in b and 'T' in b['ADC0']:
        Temperatura_ADC0 = b['ADC0']['T']

```

```

    if 'ADC0' in b and 'RH' in b['ADC0']:
        Rel_Vlaga_ADC0 = b['ADC0']['RH']

```

```

    if 'UPR2' in b and 'T' in b['UPR2']:
        Temperatura_UPR2 = b['UPR2']['T']

```

```

    if 'UPR2' in b and 'RH' in b['UPR2']:
        Rel_Vlaga_UPR2 = b['UPR2']['RH']

```

```

    if 'UPR2' in b and 'P' in b['UPR2']:
        Tlak_UPR2 = b['UPR2']['P']

```

```

    if 'SPT3' in b and 'T' in b['SPT3']:
        Temperatura_SPT3 = b['SPT3']['T']

```

```

    if 'SPT3' in b and 'RH' in b['SPT3']:
        Rel_Vlaga_SPT3 = b['SPT3']['RH']

```

```

    if 'SPT3' in b and 'P' in b['SPT3']:
        Tlak_SPT3 = b['SPT3']['P']

```

```

    if 'ERC4' in b and 'T' in b['ERC4']:

```

```

    Temperatura_ERC4 = b['ERC4']['T']

if 'ERC4' in b and 'RH' in b['ERC4']:
    Rel_Vlaga_ERC4 = b['ERC4']['RH']

if 'ERC4' in b and 'P' in b['ERC4']:
    Tlak_ERC4 = b['ERC4']['P']

if 'SHZ1' in b and 'T' in b['SHZ1']:
    Temperatura_SHZ1 = b['SHZ1']['T']

if 'SHZ1' in b and 'RH' in b['SHZ1']:
    Rel_Vlaga_SHZ1 = b['SHZ1']['RH']

if 'SHZ1' in b and 'P' in b['SHZ1']:
    Tlak_SHZ1 = b['SHZ1']['P']

if 'LOQ5' in b and 'T' in b['LOQ5']:
    Temperatura_LOQ5 = b['LOQ5']['T']

if 'LOQ5' in b and 'RH' in b['LOQ5']:
    Rel_Vlaga_LOQ5 = b['LOQ5']['RH']

if 'LOQ5' in b and 'P' in b['LOQ5']:
    Tlak_LOQ5 = b['LOQ5']['P']

if 'HJP8' in b and 'T' in b['HJP8']:
    Temperatura_HJP8 = b['HJP8']['T']

if 'HJP8' in b and 'RH' in b['HJP8']:
    Rel_Vlaga_HJP8 = b['HJP8']['RH']

if 'HJP8' in b and 'P' in b['HJP8']:
    Tlak_HJP8 = b['HJP8']['P']

if Temperatura_ADC0 and Rel_Vlaga_ADC0:
    DP = Temperatura_ADC0 - (100.0 - Rel_Vlaga_ADC0)/5

if Temperatura_UPR2 and Rel_Vlaga_UPR2:
    DP_1 = Temperatura_UPR2 - (100.0 - Rel_Vlaga_UPR2)/5

```

```

if Temperatura_SPT3 and Rel_Vlaga_SPT3:
    DP_2 = Temperatura_SPT3 - (100.0 - Rel_Vlaga_SPT3)/5

if Temperatura_ERC4 and Rel_Vlaga_ERC4:
    DP_3 = Temperatura_ERC4 - (100.0 - Rel_Vlaga_ERC4)/5

if Temperatura_SHZ1 and Rel_Vlaga_SHZ1:
    DP_4 = Temperatura_SHZ1 - (100.0 - Rel_Vlaga_SHZ1)/5

if Temperatura_LOQ5 and Rel_Vlaga_LOQ5:
    DP_5 = Temperatura_LOQ5 - (100.0 - Rel_Vlaga_LOQ5)/5

if Temperatura_HJP8 and Rel_Vlaga_HJP8:
    DP_6 = Temperatura_HJP8 - (100.0 - Rel_Vlaga_HJP8)/5

if DP_1 and DP_2 and DP_3 and DP_4 and DP_5 and DP_6 :

    DP_AV = (DP_1 + DP_2 + DP_3 + DP_4 + DP_5 + DP_5)/6

if DP < 12.5:
    Uvjeti_1 = 1
if DP >= 12.5:
    Uvjeti_1 = 0

if DP_1 < Temperatura_UPR2:
    Uvjeti_2 = 1
if DP_1 >= Temperatura_UPR2:
    Uvjeti_2 = 0

if DP_2 < Temperatura_SPT3:
    Uvjeti_3 = 1
if DP_2 >= Temperatura_SPT3:
    Uvjeti_3 = 0

if DP_3 < Temperatura_ERC4:
    Uvjeti_4 = 1
if DP_3 >= Temperatura_ERC4:
    Uvjeti_4 = 0

```

```

if DP_4 < Temperatura_SHZ1:
    Uvjeti_5 = 1
if DP_4 >= Temperatura_SHZ1:
    Uvjeti_5 = 0

if DP_5 < Temperatura_LOQ5 :
    Uvjeti_6 = 1
if DP_5 >= Temperatura_LOQ5 :
    Uvjeti_6 = 0

if DP_6 < Temperatura_HJP8:
    Uvjeti_7 = 1
if DP_6 >= Temperatura_HJP8:
    Uvjeti_7 = 0

if datetime.datetime.strptime(a, '%Y-%m-%d %H:%M:%S').timestamp() -
current_time > interval:
    time.append(current_time)
    current_time = datetime.datetime.strptime(a, '%Y-%m-%d
%H:%M:%S').timestamp()
    Fan.append(Fan_1)
    Uvjeti_ADC0.append(Uvjeti_1)
    Uvjeti_UPR2.append(Uvjeti_2)
    Uvjeti_SPT3.append(Uvjeti_3)
    Uvjeti_ERC4.append(Uvjeti_4)
    Uvjeti_SHZ1.append(Uvjeti_5)
    Uvjeti_LOQ5.append(Uvjeti_6)
    Uvjeti_HJP8.append(Uvjeti_7)
    ADC0['T'].append(Temperatura_ADC0)
    ADC0['RH'].append(Rel_Vlaga_ADC0)
    ADC0['DP'].append(DP)
    UPR2['T'].append(Temperatura_UPR2)
    UPR2['RH'].append(Rel_Vlaga_UPR2)
    UPR2['P'].append(Tlak_UPR2)
    UPR2['DP_1'].append(DP_1)
    SPT3['T'].append(Temperatura_SPT3)
    SPT3['RH'].append(Rel_Vlaga_SPT3)
    SPT3['P'].append(Tlak_SPT3)
    SPT3['DP_2'].append(DP_2)

```



```
ERC4['T'].append(Temperatura_ERC4)
ERC4['RH'].append(Rel_Vlaga_ERC4)
ERC4['P'].append(Tlak_ERC4)
ERC4['DP_3'].append(DP_3)
SHZ1['T'].append(Temperatura_SHZ1)
SHZ1['RH'].append(Rel_Vlaga_SHZ1)
SHZ1['P'].append(Tlak_SHZ1)
SHZ1['DP_4'].append(DP_4)
LOQ5['T'].append(Temperatura_SHZ1)
LOQ5['RH'].append(Rel_Vlaga_LOQ5)
LOQ5['P'].append(Tlak_LOQ5)
LOQ5['DP_5'].append(DP_5)
HJP8['T'].append(Temperatura_HJP8)
HJP8['RH'].append(Rel_Vlaga_HJP8)
HJP8['P'].append(Tlak_HJP8)
HJP8['DP_6'].append(DP_6)
SVI['DP_AV'].append(DP_AV)
```

```
if Uvjeti_1 == 1:
```

```
    Povoljno_ADC0 = Povoljno_ADC0 + interval
```

```
    Povoljno_u_satima_ADC0 = Povoljno_ADC0/3600
```

```
if Uvjeti_1 == 0:
```

```
    Nepovoljno_ADC0 = Nepovoljno_ADC0 + interval
```

```
    Nepovoljno_u_satima_ADC0 = Nepovoljno_ADC0/3600
```

```
if Uvjeti_2 == 1:
```

```
    Povoljno_UPR2 = Povoljno_UPR2 + interval
```

```
    Povoljno_u_satima_UPR2 = Povoljno_UPR2/3600
```

```
if Uvjeti_2 == 0:
```

```
    Nepovoljno_UPR2 = Nepovoljno_UPR2 + interval
```

```
    Nepovoljno_u_satima_UPR2 = Nepovoljno_UPR2/3600
```

```
if Uvjeti_3 == 1:
```

```
    Povoljno_SPT3 = Povoljno_SPT3 + interval
```

```
    Povoljno_u_satima_SPT3 = Povoljno_SPT3/3600
```

```
if Uvjeti_3 == 0:
```

```
    Nepovoljno_SPT3 = Nepovoljno_SPT3 + interval
```

```
    Nepovoljno_u_satima_SPT3 = Nepovoljno_SPT3/3600
```

```
if Uvjeti_4 == 1:
```

```

    Povoljno_ERC4 = Povoljno_ERC4 + interval
    Povoljno_u_satima_ERC4 = Povoljno_ERC4/3600
if Uvjeti_4 == 0:
    Nepovoljno_ERC4 = Nepovoljno_ERC4 + interval
    Nepovoljno_u_satima_ERC4 = Nepovoljno_ERC4/3600

if Uvjeti_5 == 1:
    Povoljno_SHZ1 = Povoljno_SHZ1 + interval
    Povoljno_u_satima_SHZ1 = Povoljno_SHZ1/3600
if Uvjeti_5 == 0:
    Nepovoljno_SHZ1 = Nepovoljno_SHZ1 + interval
    Nepovoljno_u_satima_SHZ1 = Nepovoljno_SHZ1/3600

if Uvjeti_6 == 1:
    Povoljno_LOQ5 = Povoljno_LOQ5 + interval
    Povoljno_u_satima_LOQ5 = Povoljno_LOQ5/3600
if Uvjeti_6 == 0:
    Nepovoljno_LOQ5 = Nepovoljno_LOQ5 + interval
    Nepovoljno_u_satima_LOQ5 = Nepovoljno_LOQ5/3600

if Uvjeti_7 == 1:
    Povoljno_HJP8 = Povoljno_HJP8 + interval
    Povoljno_u_satima_HJP8 = Povoljno_HJP8/3600
if Uvjeti_7 == 0:
    Nepovoljno_HJP8 = Nepovoljno_HJP8 + interval
    Nepovoljno_u_satima_HJP8 = Nepovoljno_HJP8/3600

if DP_AV < 12.5 :
    P_DP_AV = P_DP_AV + interval

if DP_AV >= 12.5 :
    N_DP_AV = N_DP_AV + interval

if Fan_1 == 1 :
    Tlacno = Tlacno + (interval/3600)

if Fan_1 == 0 :
    Ugasen = Ugasen + (interval/3600)

if Fan_1 == -1 :

```

Depresija = Depresija + (interval/3600)

```
import pyqtgraph as pg

window = pg.GraphicsLayoutWidget(title = 'test 1')
window.setBackground('w')

plot3 = window.addPlot(row=0,col=0,title = 'kretanje temperature kroz
vrijeme')
plot3.setDownsampling(auto = True)
plot3.setClipToView(clip=True)
plot3.setAxisItems({'bottom':pg.DateAxisItem()})
plot3.addLegend()
plot3.addItem(pg.PlotDataItem(time,UPR2['T'], name = 'Temperatura UPR2',
pen = '#0000FF'))
plot3.addItem(pg.PlotDataItem(time,ADC0['T'], name = 'Temperatura ADC0',
pen = '#FF0000'))
plot3.addItem(pg.PlotDataItem(time,SHZ1['T'], name = 'Temperatura SHZ1',
pen = '#008000'))
plot3.addItem(pg.PlotDataItem(time,ERC4['T'], name = 'Temperatura ERC4',
pen = '#FFFF00'))
plot3.addItem(pg.PlotDataItem(time,LOQ5['T'], name = 'Temperatura LOQ5',
pen = '#A52A2A'))
plot3.addItem(pg.PlotDataItem(time,SPT3['T'], name = 'Temperatura SPT3',
pen = '#800080'))
plot3.setLabel('left', 'Temperatura', units = '°C', color = '#000000', font
= '14pt' )
plot3.setLabel('bottom', 'vrijeme', units = 's',color = '#000000', font =
'14pt')
plot3.showGrid(x=True, y=True)
plot3.setYRange(-5, 35, padding=0)

plot4 = window.addPlot(row=1,col=0,title = ' kretanje relativna vlažnosti
kroz vrijeme ')
plot4.setDownsampling(auto = True)
plot4.setClipToView(clip=True)
```

```

plot4.setAxisItems({'bottom':pg.DateAxisItem()})
plot4.addLegend()
plot4.addItem(pg.PlotDataItem(time,UPR2['RH'], name = 'Rel.Vlaga UPR2',
pen = '#0000FF'))
plot4.addItem(pg.PlotDataItem(time,ADC0['RH'], name = 'Rel.Vlaga ADC0',
pen = '#FF0000'))
plot4.addItem(pg.PlotDataItem(time,SHZ1['RH'], name = 'Rel.Vlaga SHZ1',
pen = '#008000'))
plot4.addItem(pg.PlotDataItem(time,ERC4['RH'], name = 'Rel.Vlaga ERC4',
pen = '#FFFF00'))
plot4.addItem(pg.PlotDataItem(time,LOQ5['RH'], name = 'Rel.Vlaga LOQ5',
pen = '#A52A2A'))
plot4.addItem(pg.PlotDataItem(time,SPT3['RH'], name = 'Rel.Vlaga SPT3',
pen = '#800080'))
plot4.setLabel('left', 'Rel. Vlaga', units = '%', color = '#000000', font
= '14pt' )
plot4.setLabel('bottom', 'vrijeme', units = 's',color = '#000000', font =
'14pt')
plot4.setYRange(15, 105, padding=0)
plot4.showGrid(x=True, y=True)
plot4.setXLink(plot3)

plot5 = window.addPlot(row=4,col=0,title = 'kretanje temperature rosišta
kroz vrijeme ')
plot5.setDownsampling(auto = True)
plot5.setClipToView(clip=True)
plot5.setAxisItems({'bottom':pg.DateAxisItem()})
plot5.addLegend()
plot5.addItem(pg.PlotDataItem(time,UPR2['DP_1'], name = 'Rosište UPR2',
pen = '#0000FF'))
plot5.addItem(pg.PlotDataItem(time,ADC0['DP'], name = 'Rosište ADC0',
pen = '#FF0000'))
plot5.addItem(pg.PlotDataItem(time,SHZ1['DP_4'], name = 'Rosište SHZ1',
pen = '#008000'))
plot5.addItem(pg.PlotDataItem(time,ERC4['DP_3'], name = 'Rosište ERC4',
pen = '#FFFF00'))
plot5.addItem(pg.PlotDataItem(time,LOQ5['DP_5'], name = 'Rosište LOQ5',
pen = '#A52A2A'))
plot5.addItem(pg.PlotDataItem(time,SPT3['DP_2'], name = 'Rosište SPT3',
pen = '#800080'))

```

```

plot5.setLabel('left', 'Rosište', units = '°C', color = '#000000', font =
'14pt' )
plot5.setLabel('bottom', 'vrijeme', units = 's', color = '#000000', font =
'14pt')
plot5.setYRange(-10, 30, padding=0)
plot5.showGrid(x=True, y=True)
plot5.setXLink(plot4)
window = pg.GraphicsLayoutWidget(title = 'test')

plot1 = window.addPlot(row=0,col=0,title = 'rad ventilatora')
plot1.setDownsampling(auto = True)
plot1.setClipToView(clip=True)
plot1.addLegend()
plot1.setAxisItems({'bottom':pg.DateAxisItem()})
plot1.addItem(pg.PlotDataItem(time,Fan, name = 'Rad ventilatora', pen
='r'))
plot5 = window.addPlot(row=4,col=0,title = 'plot 5')
plot5.setDownsampling(auto = True)
plot5.setClipToView(clip=True)
plot5.setAxisItems({'bottom':pg.DateAxisItem()})
plot5.addLegend()
plot5.addItem(pg.PlotDataItem(time,Uvjeti, name = 'Rosište < 12.5', pen =
'#8C564B'))
plot5.setXLink(plot1)
import matplotlib.pyplot as plt
ugasen_dan_1 = Ugasen/2
ugasen_dan_2 = Ugasen_2/2
window.show()

plt.figure("Welcome to figure 7")
x = (1,2,3,4,5,6,)
y1,y2,y3,y4,y5,y6=
Tlacno,Depresija,ugasen_dan_1,Tlacno_2,Depresija_2,ugasen_dan_2
plt.bar(1,y1,align='center', color = 'blue')
plt.bar(2,y2,align='center', color = 'red')
plt.bar(3,y3,align = 'center', color = 'black')
plt.bar(4,y4,align = 'center', color = 'blue')
plt.bar(5,y5,align = 'center', color = 'red')
plt.bar(6,y6,align = 'center', color = 'black')
plt.xlabel('od 11 mj. 2020 do 6 mj. 2021          od 6 mj. 2021 do 11,
mj. 2021')

```

```
plt.ylabel('vrijeme (h)')
plt.title ("stanje ventilatora")
plt.legend(["tlacno","depresijski","ugasen"], loc ="upper left" )
plt.show()
```

```
plt.figure("Welcome to figure 5")
x = (1,2)
y1,y2 = ADC0_PG,ADC0_NG
plt.bar(1,y1,align='center', color ='blue')
plt.bar(2,y2,align='center', color ='red')
plt.ylabel('Povoljno vrijeme (h)')
plt.title ("ADC0")
plt.legend(["povoljno","nepovoljno", loc ="upper left" )
```