

# Karakterizacija sistema za merjenje koncentracije trdnih delcev v zraku s tipalom Sharp GP2Y1010AU0F

Žiga Patačko Koderman      Klemen Bučar

6. september 2018

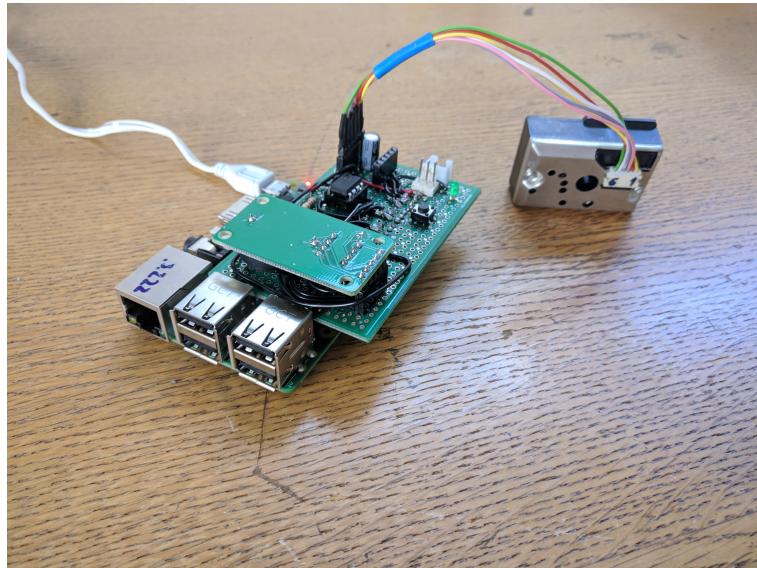
# Kazalo

<b>1</b>	<b>Uvod</b>	<b>3</b>
<b>2</b>	<b>Tipalo Sharp GP2Y1010AU0F</b>	<b>4</b>
<b>3</b>	<b>Izhod iz tipala</b>	<b>7</b>
3.1	Neenakomerno vzorčenje . . . . .	7
3.2	Velikost vzorca . . . . .	7
3.3	Svetlobne motnje iz okolice . . . . .	7
3.4	Določanje vrha . . . . .	8
3.5	Meritve v vakuumu . . . . .	8
3.6	Analiza podatkov . . . . .	8
3.6.1	Knjižnica za obdelavo podatkov . . . . .	8
3.6.2	Histogramska analiza . . . . .	10
<b>4</b>	<b>Primerjava z referenčnim tipalom</b>	<b>11</b>
<b>A</b>	<b>Izvorna koda</b>	<b>14</b>
<b>5</b>	<b>Literatura</b>	<b>15</b>

# 1 Uvod

Cilj projekta je razviti cenovno ugoden sistem za opravljanje meritev koncentracije trdnih delcev v zraku ter oceniti njegovo negotovost. Za ta namen je bilo izbrano tipalo GP2Y1010AU0F podjetja Sharp [3]. To je optični senzor, ki za delovanje izkorišča odvisnost med količino sipane svetlobe pri potovanju skozi medij in koncentracijo delcev.

prvo  
stran  
moram  
še do-  
koncati

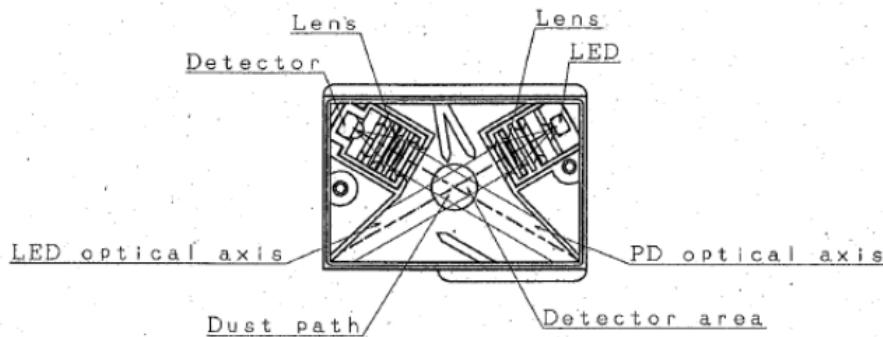


Slika 1: Raspberry Pi s priključenim tipalom Sharp GP2Y1010AU0F

## 2 Tipalo Sharp GP2Y1010AU0F

Tipalo Sharp GP2Y1010AU0F [3] sestavlja infrardeča LED, ki v pulzih emitira IR svetlobo, ter fototranzistor za detekcijo sipane svetlobe. Oba sta zaprta v ohišje in postavljena v kota ob isti stranici tako, da svetloba iz diode ne prehaja neposredno na fototranzistor (slika 2). Meritev opravimo tako, da LED vklopimo s kratkim pulzom preko RC vezja ter opazujemo odziv fototranzistorja, ki je posledica:

- odboja svetlobe od notranjih sten ohišja in
- sipanja svetlobe pri potovanju skozi medij.



Slika 2: Skica tipala GP2Y1010AU0F

Proizvajalec tipala priporoča, naj za proženje IR LED uporabimo RC vezje z  $220 \mu\text{f}$  kondenzatorjem in  $150 \Omega$  uporom [3]. Karakteristični čas takega vezja je

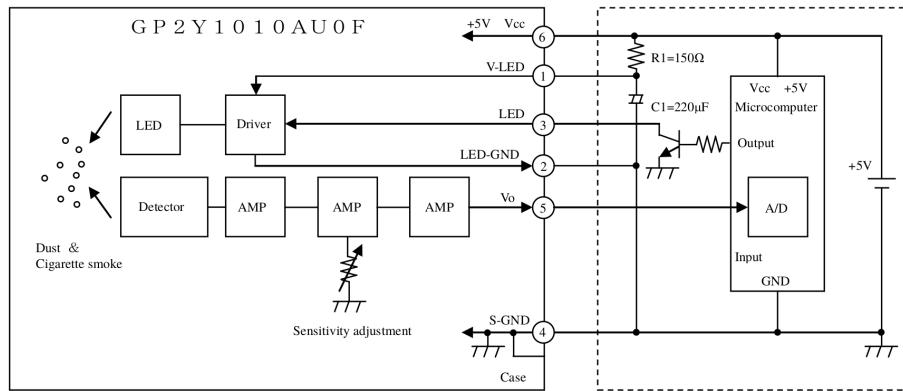
$$\tau = RC = 0,033 \text{ s}$$

Po vsakem pulzu LED moramo vezje spet napolniti, za kar porabimo nekaj karakterističnih časov. Predpostavimo, da trije karakteristični časi zadostujejo.

$$\nu = \frac{1}{t_0} \approx \frac{1}{3\tau} = 10 \text{ Hz}$$

Meritve lahko torej opravljamo s frekvenco največ 10 Hz. RC vezje nam s tem preprečuje, da bi LED preobremenili s tokom.

Po vsakem pulzu LED opazujemo odziv fototranzistorja. To je napetostni impulz, ki ga vzorčimo z 10-bitnim ADC (ang. *analog to digital*) pretvornikom MCP3002 [4], in dobimo njegovo digitalno obliko.

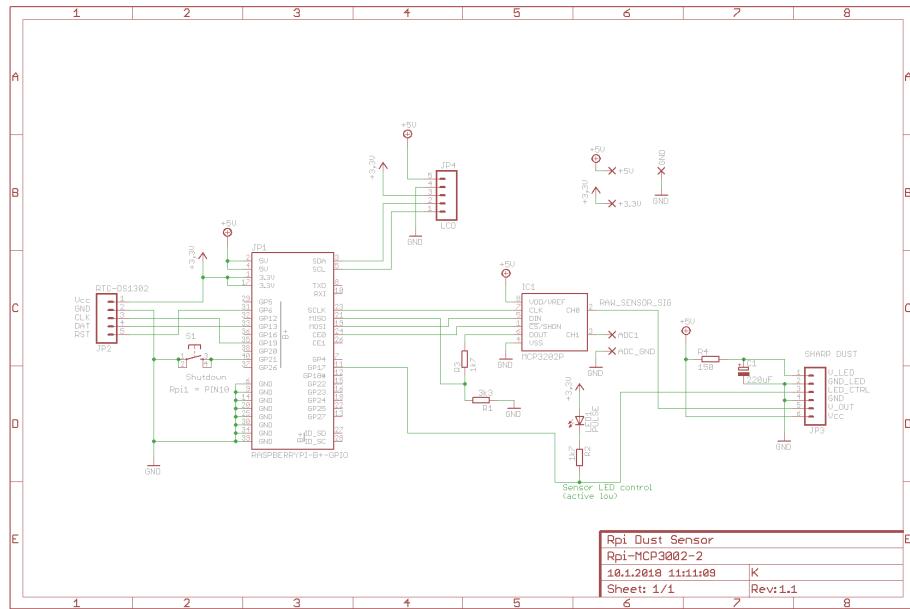


Slika 3: Primer priklopa tipala

Celoten proces, ki zajema:

- krmiljenje infrardeče LED,
  - branje pretvorjenega signala in
  - shranjevanje meritev,

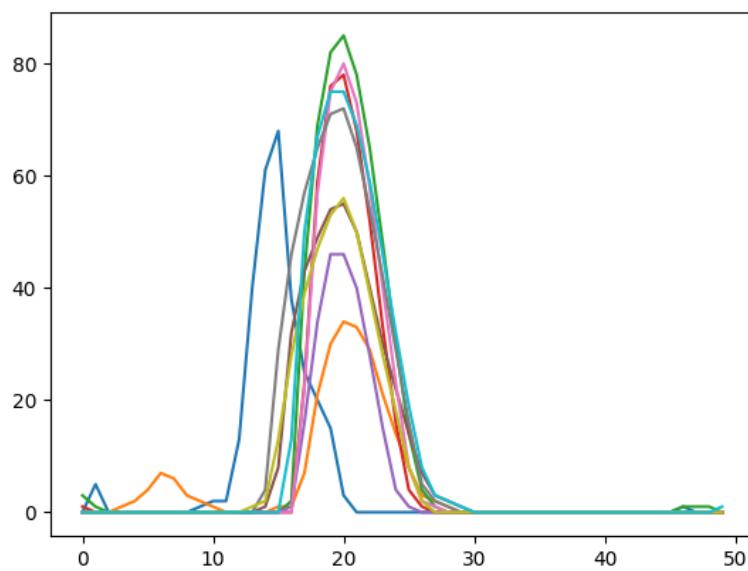
krmili mikrorodenalnik Raspberry Pi [6]. Meritvam pripisemo tudi datum in čas merjenja. V ta namen uporabimo še RTC (ang. *real time clock*) modul, saj Raspberry Pi sam tega nima.



Slika 4: Shema vezja

Prebrani pulz vzorčimo s frekvenco  $\approx 35,7\text{ kHz}$  (kolikor zmore Raspberry Pi) ter ga shranimo kot množico 50 točk, v tabelo znotraj HDF5 datoteke [2].

Proizvajalec priporoča, da napetost na fototranzistorju odčitamo 0,28 ms po vklopu LED. Kljub temu naš sistem zajema odziv fototranzistorja 1,4 ms od vklopa LED. Čas 0,28 ms sovpada z vrhovi izmerjenih pulzov, tako da predvidevamo, da v resnici iščemo vrh pulza.



Slika 5: Primer 10 odzivov fototranzistorja pri vzorčenju 50 točk

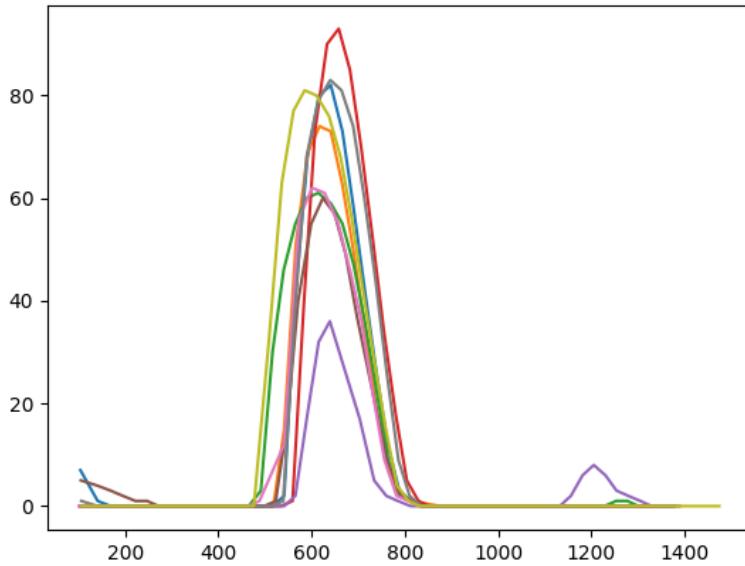
S slike 5 je razvidno, da vzorci pulzov med seboj niso preveč podobni. Vrhovi so zelo razpršeni in odstopajo za  $\approx \pm 40\%$ .

### 3 Izhod iz tipala

Zaradi neuporabnosti obstoječih meritev se odločimo za sistematično analizo in nadgradnjo obstoječega sistema za zajem in analizo vzorcev.

#### 3.1 Neenakomerno vzorčenje

Zaradi variabilne obremenitve procesorja na krmilniku Raspberry Pi dobi proces za zajemanje meritve spremenljivo količino procesorskega časa. Točk na pulzu torej ne odčitavamo z enakomerno frekvenco. Odstopanja odpravimo tako, da vsaki točki dodamo časovni žig.



Slika 6: Pulzi po časovni poravnani izmerjenih točk v  $\mu$ s

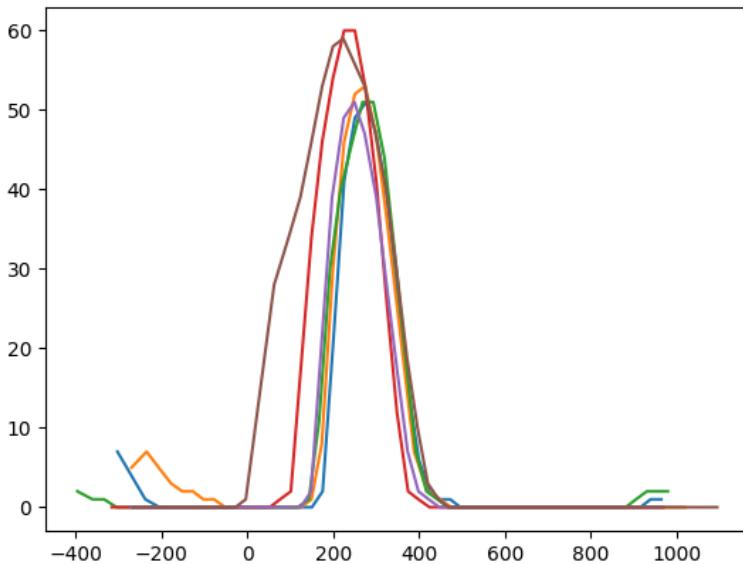
Ko točke poravnamo relativno na čas vžiga LED, opazimo precej lepšo poravnavo vrhov (slika 6).

#### 3.2 Velikost vzorca

Pri opravljanju meritve ne poznamo natančnega pretoka zraka skozi tipalo. Zanašamo se na konvekcijsko gibanje zraka. Da preverimo, ali to zadostuje za potrebe meritve, pretok prisilno povečamo z ventilatorjem ter opazujemo morebitne spremembe. Ne opazimo nobenih sprememb ter sklepamo, da je konvekcijski pretok zraka dovoljen.

#### 3.3 Svetlobne motnje iz okolice

Ohišje tipala je odprto, da lahko skozenj teče zrak. To pa fototranzistor v tipalu izpostavi zunanjim svetlobnim virom. Ti lahko na meritve vplivajo v obliki šuma ali sistematične napake (na primer razlika med dnevom in nočjo). To preverimo s postavitvijo tipala v popolno temo. Opazimo, da so vrhovi pulzov poravnani bistveno bolje (slika 7). Meritve od tu naprej opravljamo v temi.



Slika 7: Pulzi, izmerjeni v temi v  $\mu$ s

### 3.4 Določanje vrha

Do zdaj smo pri zajemanju podatkov za vrh pulza vzeli kar najvišjo izmerjeno točko. Ta pa zaradi premajhne frekvence vzorčenja nikoli ne sovpada popolnoma s pravim vrhom. Napako odpravimo z napenjanjem parabole na najvišjih nekaj točk. Vsak pulz zajema približno med 10 in 20 točk, tako da se odločimo za napenjanje parabole na najvišjih 7.

### 3.5 Meritve v vakuumu

Navkljub vsem izboljšavam meritve ostajajo polne šuma. Da bi to kvantificirali, postavimo tipalo v vakuum ter popolno temo. S tem želimo odstraniti vse zunanje dejavnike ter izmeriti izključno odboj svetlobe od ohišja. Napaka meritve v tem primeru prihaja iz negotovosti pri odčitavanju ter neenakomernih pulzov LED.

Razlike med meritvami na prostem in v  $15\text{mbar}$  vakuumu ne zaznamo. Sklepamo, da potrebujemo še nižji pritisk.

### 3.6 Analiza podatkov

Analiza podatkov poteka na ločenem računalniku in ni del naprave za opravljanje meritev. Obstojeci sistem za obdelavo meritev povpreči višine vrhov pulzov v želenem časovnem obdobju (nekaj 10 minut). Rezultate skupaj z meritvami, opravljenimi z merilnikom Grimm mini-LAS, nariše na graf. Posebne korelacije ne opazimo. Da bi to kvantificirali, korelacijo tudi izračunamo. Ta je majhna - za večino en dan trajajočih meritev je njena vrednost pod 0,5, kar ni uporabno.

#### 3.6.1 Knjižnica za obdelavo podatkov

Za hitrejšo in lažjo analizo podatkov je bilo več manjših programov združenih v Python knjižnico po imenu [Analysis](#). Ta je zelo prilagodljiva in uporabniku omogoča nalaganje podatkov o pulzih

iz hdf5 [2] datotek in risanje grafov s samo nekaj ukazi.

Obdelava velike količine podatkov (včasih tudi za več dni) je zelo dolgotrajna. Knjižnica [Analysis](#) je zato optimizirana z uporabo knjižnice [NumPy](#) za hitrejše računanje. Da bi prihranili še več časa, je obdelava podatkov razbita na dva dela:

- predpripričavo podatkov (iskanje vrha pulzov, povprečenje na želenem intervalu, izločanje neuporabnih pulzov itd.) ter
- analizo (risanje grafov, histogramov ter računanje korelacijs).

Predpripričava podatkov ostaja večinoma nespremenjena, zato je podatke po prvem delu smiselno shraniti na disk. V našem primeru knjižnica shrani podatke v formatu json [5]. Primer take izvorne kode je viden na sliki 8.

Analiza že predpripričavljenih podatkov pa se pogosto spreminja oz. izpopolnjuje. Uporabnik lahko prvi del poganja samo, kadar je to zares nujno potrebno, potem pa pri analizi že pripravljene podatke preprosto nalaga z diska. Primer nalaganja že obdelanih datotek ter risanja grafov je viden na sliki 9.

```
import analysis
import json_tricks

# seznam datotek
rbpi3_files = [
    'meritev004.h5',
    'meritev005.h5',
]

# nalaganje datotek
rbpi3 = analysis.from_sharp_dust(rbpi3_files, 6000,
                                   show_std_dev=False)
rbpi3.set_measurement_data('RbPi3', 'b', y_lim=(71.5, 75.5))
measurements['rbpi3'] = rbpi3

# shranjevanje grobo obdelanih podatkov
with open('dump.json', 'w') as f:
    f.write(json_tricks.dumps(measurements))
```

Slika 8: Primer izvorne kode za nalaganje, obdelavo podatkov ter shranjevanje v datoteko dump.json

```

import analysis
import json_tricks

# nalaganje podatkov
with open( 'dump.json' , 'r' ) as f:
    measurements = json_tricks.loads( f.read() )

# plot risanje grafov
analysis.plot_average_over_time(
    [ measurements[ '0.28um' ] ,
    measurements[ '0.40um' ] ,
    measurements[ 'rbpi3' ] ]
)

```

Slika 9: Primer izvirne kode za nalagane obdelanih podatkov iz datoteke dump.json ter risanje grafov

### 3.6.2 Histogramska analiza

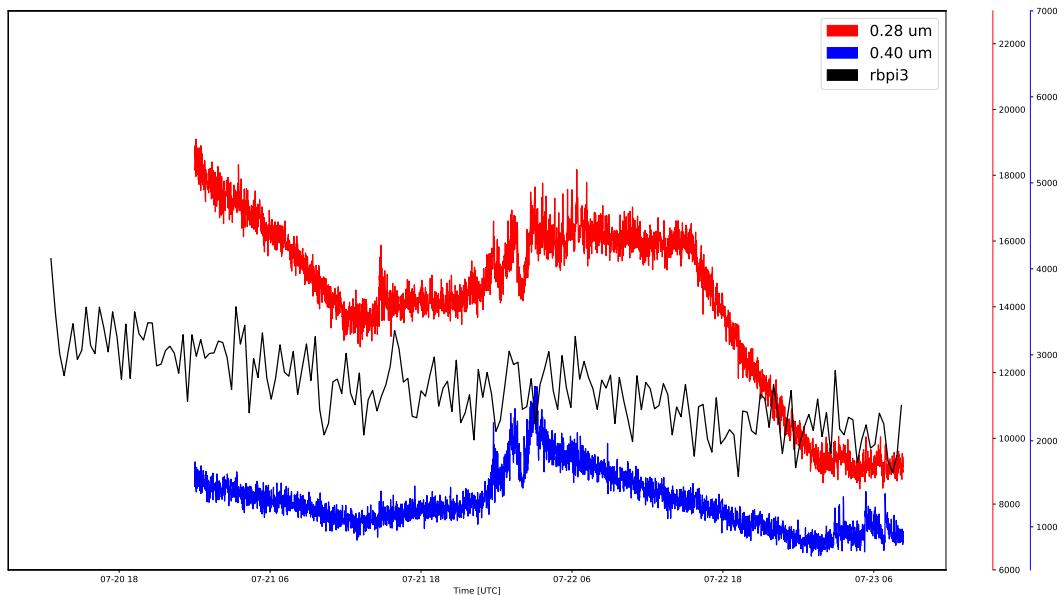
Odziv fototranzistorja na pulz LED ni odvisen samo od gostote trdnih delcev v zraku. Velikost delca igra veliko vlogo. Višina pulza je verjetneje povezana z velikostjo zaznanih delcev kot z gostoto delcev v zraku. To pomeni, da lahko preštejemo pulze različnih višin znotraj določenega intervala in jih primerjamo z meritvami različno velikih delcev referenčnega tipala.

Program šteje pulze v spreminjačem se intervalu ter jih primerja z meritvami različno velikih delcev. Tako poišče intervale, ki najbolj ustreza posameznim velikostim delcev iz meritev referenčnega tipala. Rezultati so razvidni iz tabele 1.

## 4 Primerjava z referenčnim tipalom

Da karakteriziramo natančnost merilnega sistema, mertive primerjamo z referenčnim tipalom Grimm Mini-LAS 11-R [1]. To tipalo meri koncentracije trdnih delcev posamzenih velikosti v zraku.

Korelacije med osnovnimi meritvami in meritvami referenčnega tipala se gibljejo okoli 0,7. Primer grafa je viden na sliki 10.

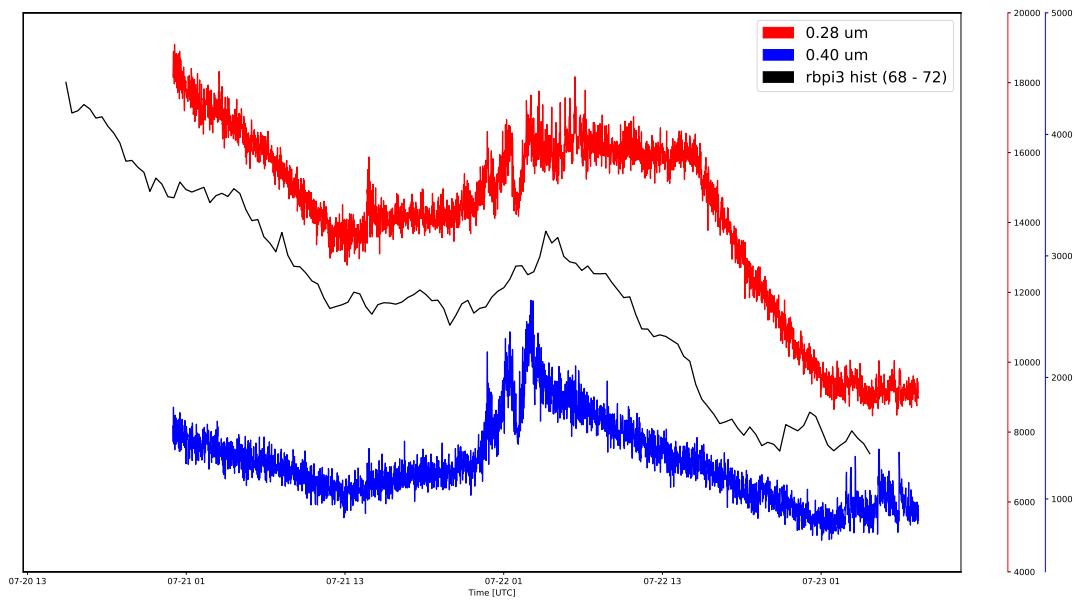


Slika 10: Primerjava meritev z referenčnim tipalom

Za analizo uporabimo še prej opisano metodo štetja vrhov na posameznem intervalu. Program izpiše tabelo 1, iz katere je razvidno, da z štetjem vrhov na pravem intervalu dobimo korelacije, ki presegajo 0,9. To potrjuje tezo Meritve tipala Sharp GP2Y1010AU0F najbolje korelirajo z koncentracijami delcev velikosti med 0,25 in 0,50  $\mu\text{m}$ .

velikost [ $\mu m$ ]	korelacija	interval
0,25	0,94	68 - 72
0,28	0,94	68 - 72
0,30	0,93	68 - 72
0,35	0,93	76 - 84
0,40	0,96	76 - 84
0,45	0,91	76 - 84
0,50	0,91	80
0,58	0,74	80
0,65	0,39	80
0,70	0,30	48
0,80	0,34	48
1,00	0,44	48
1,30	0,50	52
1,60	0,59	52
2,00	0,70	52
2,50	0,71	52
3,00	0,66	52
3,50	0,67	64
4,00	0,68	64
5,00	0,68	64
6,50	0,68	64
7,50	0,68	64
8,50	0,68	64
10,00	0,68	64
12,50	0,67	64
15,00	0,66	64
17,50	0,66	64
20,00	0,66	64
25,00	0,66	64
30,00	0,66	64

Tabela 1: Intervali korelaciij pri primerjavi histogramske analize podatkov z referenčnim tipalom



Slika 11: Primerjava meritev po histogramski analizi z referenčnim tipalom

## Dodatek A Izvorna koda

---

potrebujem  
izvorno  
kodo  
kot  
prilogo?

## 5 Literatura

- [1] Grimm. *Grimm Portable Laser Aerosol Spectrometer*. URL: [http://wiki.grimmaerosol.de/images/b/b8/D\\_E\\_11-E\\_Mini-LAS\\_rev1p1.pdf](http://wiki.grimmaerosol.de/images/b/b8/D_E_11-E_Mini-LAS_rev1p1.pdf) (pridobljeno 2018).
- [2] HDF Group. *HDF5 File Format*. URL: <https://support.hdfgroup.org/HDF5/doc1.8/H5.intro.html> (pridobljeno 2018).
- [3] Sharp. *GP2Y1010AU0F Application Note*. URL: [http://www.sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y1010au\\_appl\\_e.pdf](http://www.sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y1010au_appl_e.pdf) (pridobljeno 2018).
- [4] Microchip Technology. *MCP3002*. URL: <http://ww1.microchip.com/downloads/en/DeviceDoc/21294E.pdf> (pridobljeno 2018).
- [5] The Free Encyclopedia Wikipedia. *JSON*. URL: <https://en.wikipedia.org/wiki/JSON> (pridobljeno 2018).
- [6] The Free Encyclopedia Wikipedia. *Raspberry Pi*. URL: [https://en.wikipedia.org/wiki/Raspberry\\_Pi](https://en.wikipedia.org/wiki/Raspberry_Pi) (pridobljeno 2018).