



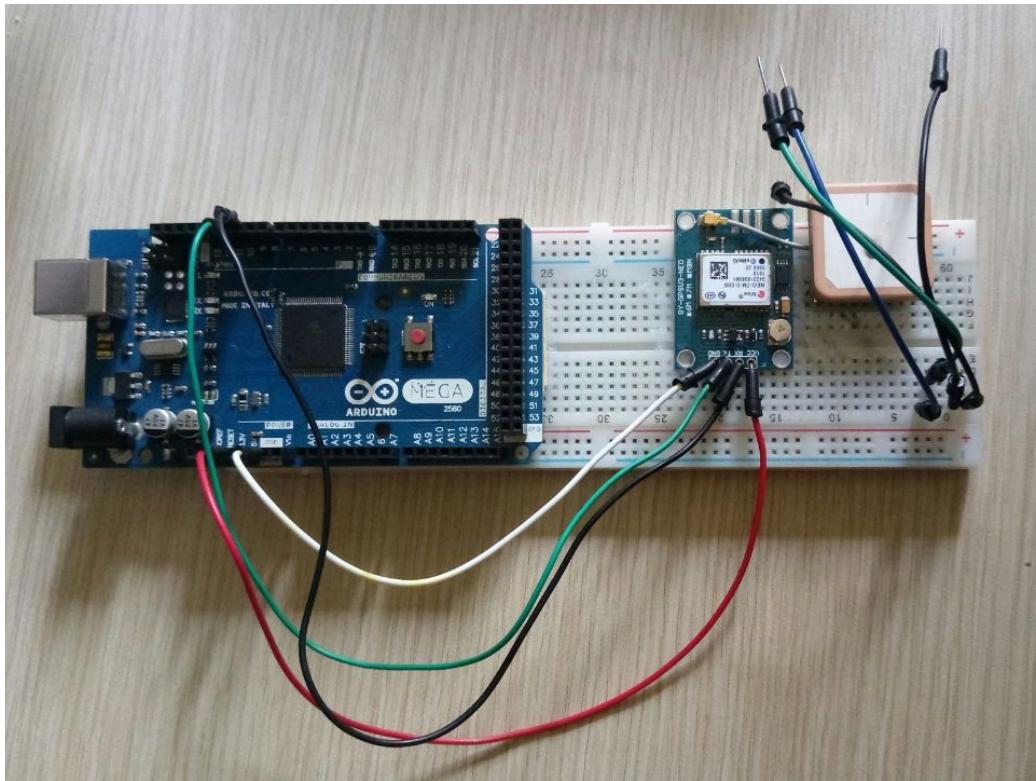
آزمایشگاه سخت افزار دکتر فصحتی

گروه ۳

اعضای گروه
محمد جمشیدی
مهری فرزادی

گزارش کلی

معماری مدار



اجزای پروژه

- سخت افزار
- نرم افزار

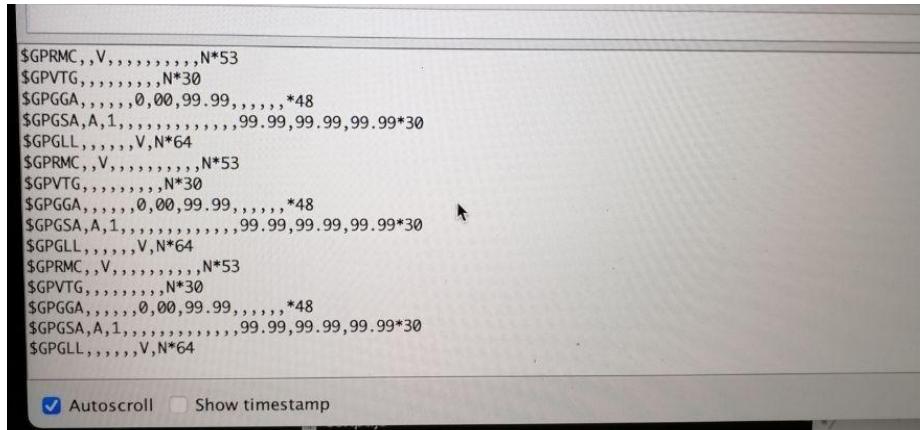
سخت افزار

قطعات بکار رفته

- آردوینو مگا
- مازول GPS
- هشت سیم جامپر
- برد برد

چالش بخش سخت افزاری

ابتدا سعی کردیم مژول جی پی اس را راه اندازی کنیم و دیتای آن را جمع آوری کرکه این مژول اهمیت زیادی برای پروژه دارد. ابتدا سعی کردیم جی پی اس را به پین های سریال آردوینو متصل کنیم تا بتوانیم دیتا های خروجی این سنسور را بدون کد خاصی در نمایشگر سریال مشاهده کنیم. سنسور جی پی اس به محض اینکه روشن می شود تلاش می کند تا خود را کالیبره کند و به ماهواره ها متصل شود و وقتی که به ماهواره ها متصل شد چراغ led روی این سنسور شروع به زدن چشمک در هر ثانیه می کند. البته این فرایند متصل شدن به ماهواره تقریبا ۳۰ ثانیه به طول می آنجامد. اما مشکلی که وجود داشت این بود که این سنسور در محیط های بسته نمی تواند به ماهواره متصل شود و در این شرایط led آن شروع به چشمک زدن نمی کند. همچنین دیتا درستی از سنسور در پین TX آن قرار نمی گیرد و عمل سنسور کار نمی کند و دیتای اشتباهی در نمایشگر سریال آردوینو مشاهده می شود.



برای پیدا کردن علت این مشکل زمان زیادی صرف کردیم. ابتدا احتمال دادیم که مشکل از نحوه بسته شدن مدار باشد و برای بررسی این مشکل پین های مختلف آردوینو را به سنسور متصل کردیم اما مشکل برطرف نشد. سپس از صحت سیم ها اطمینان حاصل کردیم. در نهایت به طور اتفاقی سنسور را جابه جا کردیم و وقتی کنار پنجره بردم، سنسور شروع به چشمک زدن کرد و دیتای خروجی در نمایشگر سریال آردوینو به نمایش درآمد.

```
$GPGSA,A,3,28,13,17,30,07,01,,,,,,1.70,1.43,0.92*02
$GPGSV,3,1,12,17,75,198,37,28,57,017,13,30,57,113,30,13,48,276,36*7D
$GPGSV,3,2,12,19,45,208,22,01,24,071,19,11,24,053,15,07,23,125,20*77
$GPGSV,3,3,12,15,21,303,16,18,11,047,,06,02,179,,35,,,*4D
$GPRMC,220739.000,A,3244.8433,N,11711.4519,W,0.30,206.10,240319,,,A*79
$GPVTG,206.10,T,,,M,0.30,N,0.56,K,A*38
$GPGGA,220740.000,3244.8432,N,11711.4519,W,1,06,1.43,57.0,M,-35.9,M,,*6F
$GPGSA,A,3,28,13,17,30,07,01,,,,,,1.70,1.43,0.92*02
$GPRMC,220740.000,A,3244.8432,N,11711.4519,W,0.12,102.83,240319,,,A*7B
$GPVTG,102.83,T,,,M,0.12,N,0.22,K,A*36
$GPGGA,220741.000,3244.8432,N,11711.4519,W,1,06,1.43,57.0,M,-35.9,M,,*6E
$GPGSA,A,3,28,13,17,30,07,01,,,,,,1.70,1.43,0.92*02
```

تبدیل دیتای سنسور به طول و عرض جغرافیایی

مشکل بعدی این بود که دیتای خروجی سنسور به فرمت خام بود و ما نیاز به طول و عرض جغرافیایی داشتیم تا بتوانیم از این دیتا برای تست کردن الگوییتم کالمن استفاده کنیم.
در این مرحله سعی کردیم دیتای خام سنسور را توسط آردینو به فرمت دلخواه خودمان در بیاوریم برای این منظور از لایبرری **TinyGPS++** استفاده کردیم و دیتا را به فرمت مناسب تبدیل کردیم.
لازم به ذکر است که دیتای سنسور توسط آردینو به طول و عرض جغرافیایی تبدیل شده است. در واقع کد بر روی آردینو اجرا شده است.
کد مذکور در مخزن گیت پروژه قرار داده شده است.

```
GPS GY-GPS6MV2
---Buscando señal---

Latitud/Longitud: 7.06685, -73.07704
Fecha: 23/6/2018 Hora: 2:29:52.0
Altitud (metros): 1000000.00
Rumbo (grados): 0.00
Velocidad(kmph): 0.19
Satélites: 255

Latitud/Longitud: 7.06685, -73.07704
Fecha: 23/6/2018 Hora: 2:29:53.0
Altitud (metros): 1000000.00
Rumbo (grados): 0.00
Velocidad(kmph): 0.69
Satélites: 255

Latitud/Longitud: 7.06685, -73.07706
Fecha: 23/6/2018 Hora: 2:30:3.0
Altitud (metros): 1000000.00
Rumbo (grados): 0.00
Velocidad(kmph): 0.07
Satélites: 255

Latitud/Longitud: 7.06686, -73.07706
Fecha: 23/6/2018 Hora: 2:30:4.0
Altitud (metros): 1000000.00
Rumbo (grados): 0.00
Velocidad(kmph): 0.48
Satélites: 255

Latitud/Longitud: 7.06686, -73.07707
Fecha: 23/6/2018 Hora: 2:30:5.0
Altitud (metros): 1000000.00
Rumbo (grados): 0.00
Velocidad(kmph): 0.31
Satélites: 255

 Autoscroll   
```

Fecha: 20/8/2022 Hora: 15:21:28.0

Altitud (metros): 1231.00

Rumbo (grados): 285.01

Velocidad(kmph): 1.37

Satélites: 5

Latitud/Longitud: 35.71076, 51.34265

Fecha: 20/8/2022 Hora: 15:21:29.0

Altitud (metros): 1231.00

Rumbo (grados): 285.01

Velocidad(kmph): 2.85

Satélites: 5

Latitud/Longitud: 35.71075, 51.34265

Fecha: 20/8/2022 Hora: 15:21:30.0

Altitud (metros): 1231.00

Rumbo (grados): 285.01

Velocidad(kmph): 3.37

Satélites: 5

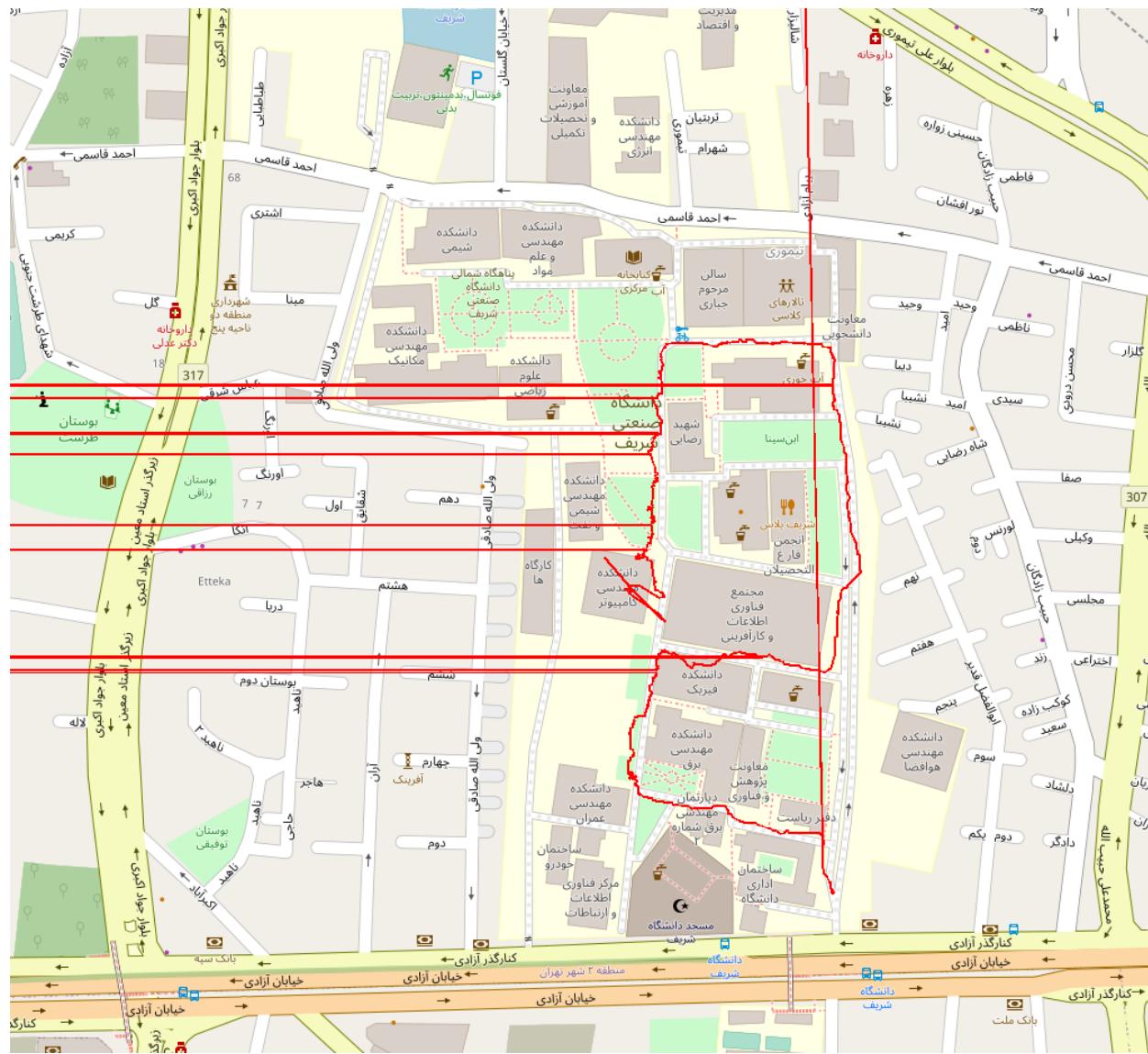
ذخیره دیتای GPS

با توجه به مسائل ذکر شده درباره مشکلات قطع و وصل شدن سنسور GPS، گرفتن دیتا به صورت آنلاین از GPS و پردازش آن در لحظه توسط الگوریتم کالمن، کاری دشوار بود و فرآیند توسعه پروژه را کند می‌کرد. برای حل این مشکل، با هماهنگی استاد درس، دریافت دیتا از سنسور و پردازش آن را از هم جدا کردیم. به این صورت که یک مسیر مشخص و به اندازه کافی بلند در دانشگاه را انتخاب کردیم و دیتای خروجی سنسور را در یک فایل متغیر ذخیره کردیم تا بخش نرم افزاری پروژه به صورت آفلاین از این دیتا استفاده کند. همچنین برای تست کردن بخش نرم افزاری نیاز نباشد که بخش سخت افزاری هم فعال باشد که این امر موجب افزایش سرعت دولوب پروژه می‌شود.

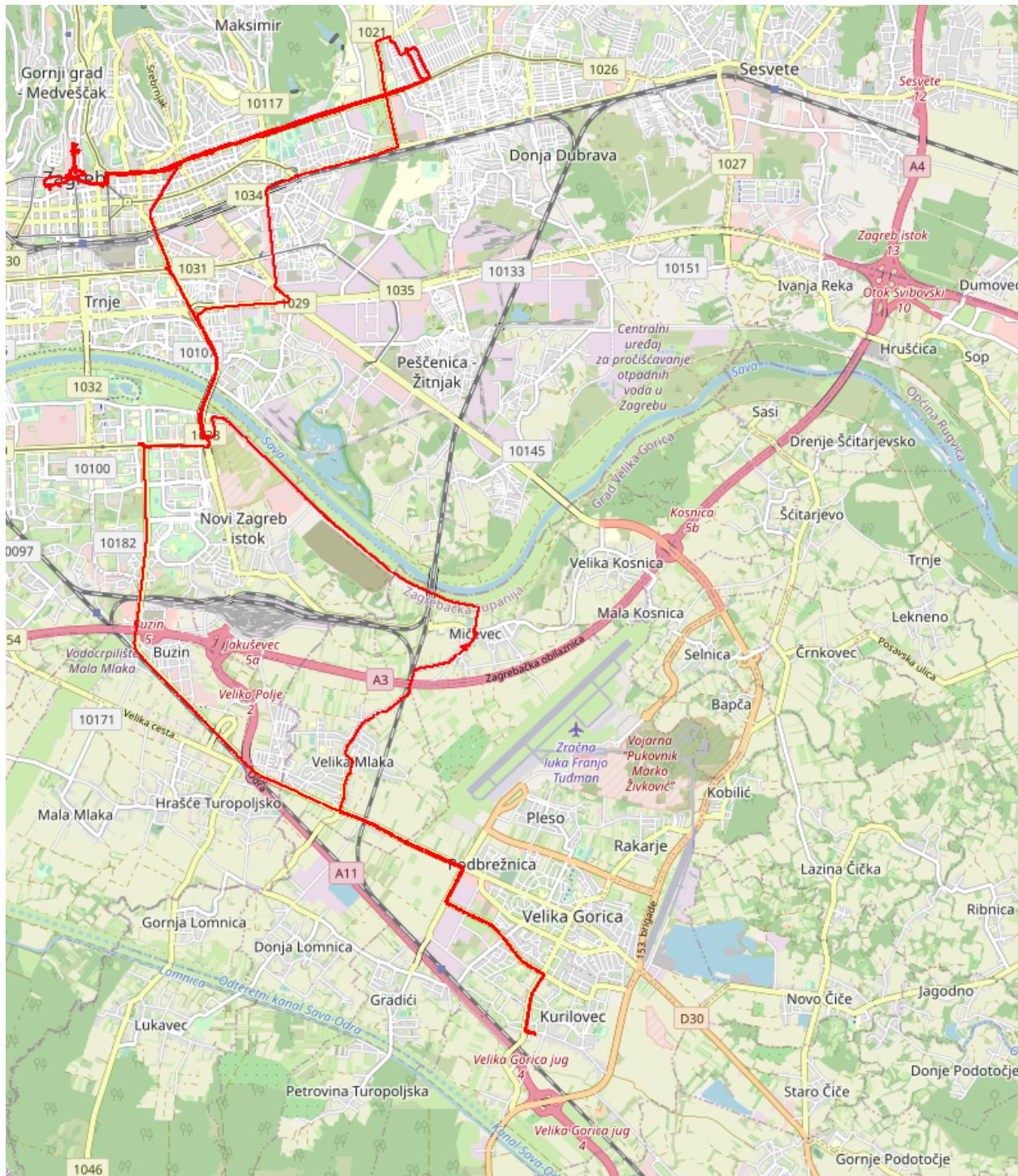
بخش نرم افزاری

بخش نرم افزاری از یک Jupyter Notebook تشکیل شده است که ابتدا دیتای سنسور را از فایل می‌خواند و به فرمت مناسب برای استفاده الگوریتم کالمن تبدیل می‌کند.

سپس طول و عرض جغرافیایی را روی نقشه متضاظر کردیم تا مسیر طی شده، بهتر مشخص شود. بعد از ویژوالیز کردن داده‌های سنسور متوجه میزان خطای زیاد سنسور شدیم. همچنین در حین جمع کردن داده، برای لحظاتی کوتاه سنسور ارتباط خود را با سیستم ماهواره‌ای از دست میداد و در آن لحظات دیتای سنسور با خطای شدیدی همراه می‌شد. این میزان خطای زیر قابل مشاهده است:



با در اختیار داشتن داده‌ای که با این میزان خطأ همراه است، طبیعتاً نمی‌توانستیم الگوریتم کالمن را تست و ارزیابی کنیم. برای رفع این مشکل، علاوه بر ارزیابی داده‌های سنسور، از یک Data Set آمده نیز استفاده کردیم که شامل ۳۵۰۰۰ نقطه بود. تصویر این دیتاست بر روی نقشه به شکل زیر است



نحوه مپ کردن دیتای سنسور به نقشه

برای مپ کردن نقاط، نیاز است داده‌های سنسور را به ابعاد عکس گرفته شده از نقشه تبدیل کنیم. به این شکل که مبدا مختصات را سمت راست پایین عکس در نظر میگیریم و سمت چپ بالا برابر با ابعاد عکس درنظر میگیریم. سپس طول و

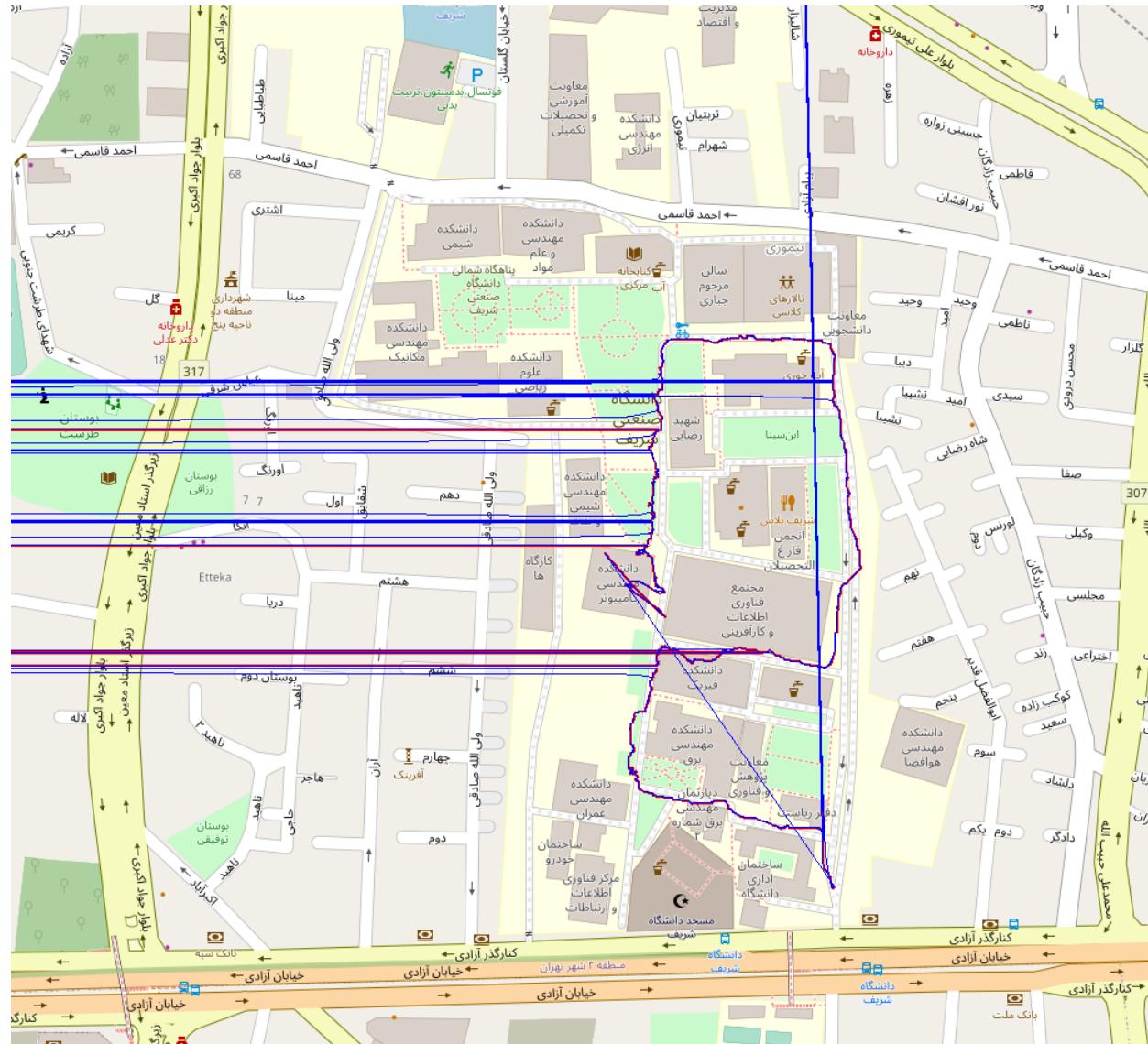
عرض جغرافیایی این دو نقطه را بدست می‌آوریم از این طریق رابطه نقاط نقشه و نقاط عکس را محاسبه می‌کنیم. در مرحله بعد با استفاده از این رابطه، داده‌های سنسور را بر روی عکس می‌نگاریم.

الگوریتم کالمن

الگوریتم کالمن از دو بخش `predict`, `update` تشکیل شده است. به این شکل که بعد از دیدن هر نقطه، ابتدا بخش `update` اجرا می‌شود و در این بخش پارامترهای الگوریتم با توجه به دیتای جدید، بهبود داده می‌شود. با استفاده از بخش `predict` الگوریتم کالمن می‌توانیم نقطه بعدی کاربر را پیش‌بینی کنیم.

همچنین این الگوریتم را می‌توان در `order` های مختلف اجرا کرد. الگوریتم کالمن با اردر صفر مربوط به حالتی است که فقط از مکان نقاط برای پیش‌بینی استفاده می‌کند. الگوریتم کالمن با اردر یک، علاوه بر مکان نقاط از سرعت حرکت کاربر نیز استفاده می‌کند و مدلی پیچیده‌تر نسبت به الگوریتم کالمن با اردر صفر دارد. همچنین الگوریتم کالمن با اردر دو، علاوه بر مکان و سرعت کاربر، شتاب حرکت کاربر را نیز در مدل خود دخیل می‌کند تا بتواند پیش‌بینی بهتری داشته باشد. اما با توجه به اینکه سنسور فقط مکان نقاط را اندازه‌گیری می‌کند، در این پروژه فقط امکان اجرای الگوریتم کالمن با اردر صفر را دارا بودیم.

در واقع روند تست الگوریتم به این شکل بود که داده‌های سنسور را به ترتیب به بخش `update` الگوریتم ارسال می‌کردیم و سپس با استفاده از تابع `predict` مکان بعدی را پیش‌بینی می‌کردیم. سپس با جمع آوری نقاط پیش‌بینی شده و مپ کردن این نقاط بر روی نقشه توانستیم میزان خطای نقاط پیش‌بینی شده توسط الگوریتم و نقاط واقعی را ویژوالایز کنیم که در شکل زیر نتیجه کار قابل مشاهده است.

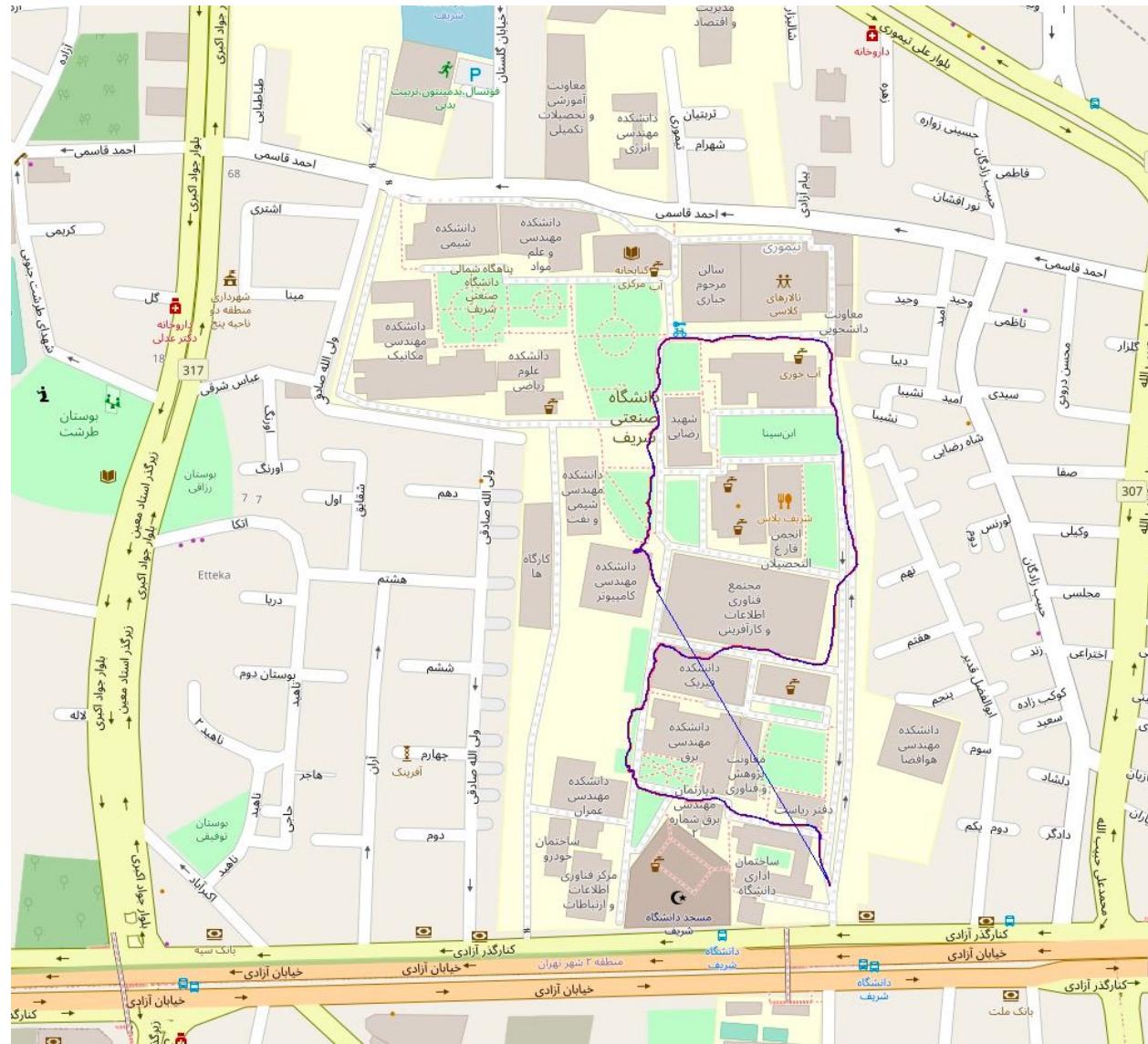


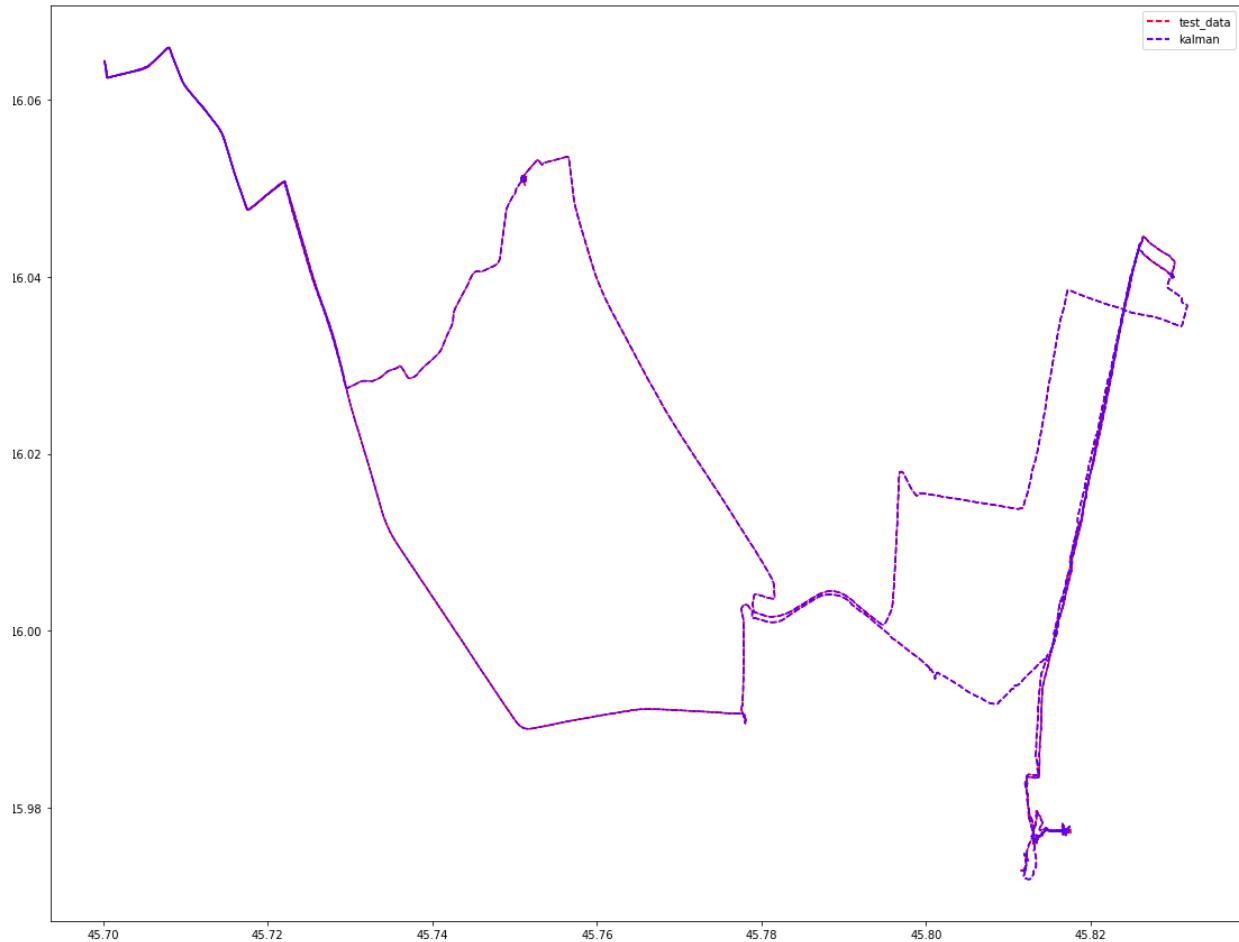
نکته برجسته در تصویر فوق این است که الگوریتم کالمن نتوانسته است نویز و خطای حاصل از سنسور را تشخیص دهد و این خطرا هندل کن. نقاط واقعی با رنگ قرمز و نقاط پیش‌بینی شده توسط الگوریتم کالمن با رنگ آبی مشخص شده‌اند. علت این عملکرد این است که زمانی که سنسور ارتباط خود با ماهواره را از دست می‌داد فقط یک نقطه ثبت نمی‌کرد بلکه در این حین ممکن بود ۱۰ نقطه اشتباه متواالی را ثبت کند (بسته به زمان کانکت شدن مجدد این عدد متغیر است).

راه حل بکار گرفته شده برای حل این مشکل، استفاده از روشی به اسم **windowing** است. به این شکل که یک پنجره با طول ثابت برای مثال با طول ۱۰ را در نظر می‌گیریم و این پنجره را روی دادگان حرکت می‌دهیم و در هر مرحله میانگین اعدادی که در این پنجره قرار می‌گیرند را بعنوان یک داده نرمالایز شده در نظر می‌گیریم. با استفاده از این روش اثر داده‌های پرت را کم می‌کنیم. چالش موجود بر سر انتخاب مقدار بهینه برای سایز پنجره است. با تست کردن مقدارهای مختلف برای سایز پنجره، عدد ۵ بعنوان سایز نهایی پنجره انتخاب شده است.

با اعمال کردن روش ذکر شده برای نرم‌الایز کردن دادگان و سپس مپ کردن نقاط دیتا است بر روی نقشه نتیجه زیر حاصل شد که همانطور که کاملاً مشخص است، نقاط پرت و نویزی به طور کامل اثر خود را از دست داده‌اند و یک مسیر مشخص ساخته شده است.







محاسبه خط

برای محاسبه خط، ابتدا میانگین قدر مطلق نقاط واقعی و نقاط پیش‌بینی شده توسط الگوریتم کالمن را محاسبه کردیم و سپس آن را ویژولایز کردیم. اما چون نقاط از جنس طول و عرض جغرافیایی هستند میزان خطابسیار کوچک می‌شود اما این به معنی میزان خطای واقعی کم نیست. زیرا اختلاف اندازه در طول و عرض جغرافیایی به معنی اختلاف چند ۱۰۰ متری بر روی نقشه است. برای رفع این مشکل ابتدا اختلاف نقاط واقعی با نقاط پیش‌بینی شده متناظر را به متر تبدیل کردیم و سپس از آن میانگین گرفتیم که نمودار زیر حاصل شد. همانطور که واضح است میانگین خطابسیار ۳ متر است.

