



دانشگاه تهران
دانشکده‌ی مهندسی برق و کامپیوتر



گزارش نهایی پروژه سیستم‌های سایبر-فیزیکی طراحی سیستم پایش موقعیت و شتاب خودرو بر بستر شبکه GPRS

گروه:

محمد رضا بخشایش و امین باهنر و علی پرویزی

تاریخ:

مرداد ۱۴۰۴

استاد:

دکتر مهدی مدرسی و دکتر مهدی

کارگهی

۱۴۰۳-۱۴۰۴

فهرست مطالب

- ۱- مقدمه ۳
- ۲- معرفی پلتفرم و ابزارهای استفاده شده ۴
- ۳- تغییرات نسبت به فاز پروپوزال ۵
- ۴- نزدیک ترین نمونه های مشابه ۵
- ۵- مبانی فنی پروژه ۶
- ۶- جزئیات پیاده سازی ۱۰
- ۷- آزمون، ارزیابی و مقایسه عملکرد ۱۱
- ۸- پیوست های فنی ۱۲
- ۹- مراجع Error! Bookmark not defined.

1- مقدمه

پایش موقعیت و وضعیت دینامیکی خودروها، از جمله موقعیت مکانی، سرعت، شتاب، و جهت حرکت در طول زمان، نقش مهمی در مدیریت ناوگان حمل و نقل، ردیابی دارایی‌ها، و بهینه‌سازی عملیات لجستیک ایفا می‌کند. این فناوری امکان نظارت لحظه‌ای بر عملکرد خودروها را برای شرکت‌های ارائه‌دهنده خدمات حمل و نقل فراهم می‌کند، اطلاعاتی ارزشمند در اختیار رانندگان قرار می‌دهد، و قابلیت ردیابی دقیق را در سناریوهای مختلف، از جمله مدیریت ناوگان یا ردیابی وسایل نقلیه سرقتی، میسر می‌سازد. اهمیت این سیستم‌ها در مناطق با پوشش شبکه محدود، مانند بسیاری از نقاط ایران که شبکه‌های ۲G و ۴G دسترس نیستند یا شبکه GPRS با پایداری پایین عمل می‌کند، دوچندان است. در چنین شرایطی، نیاز به سیستمی با تأخیر کم، قابل اعتماد، و مقرون به صرفه برای پایش خودروها به شدت احساس می‌شود.

روش‌های سنتی پایش خودرو، که اغلب به دستگاه‌های GPS تجاری وابسته هستند، با چالش‌هایی نظیر هزینه‌های بالا، وابستگی به زیرساخت‌های پیچیده، و عدم انعطاف‌پذیری در جمع‌آوری و ارسال داده‌های متنوع مانند شتاب یا دمای موتور مواجه‌اند. این سیستم‌ها در شبکه‌های ناپایدار، مانند شبکه GPRS در ایران، عملکرد مطلوبی ندارند و قادر به تأمین نیازهای پایش با تأخیر کم نیستند. این پروژه سیستمی نهفته را توسعه داده است که با استفاده از سخت‌افزارهای مقرون به صرفه نظیر میکروکنترلرها و ماژول‌های ارتباطی، داده‌های کلیدی خودرو شامل موقعیت، سرعت، شتاب و جهت را جمع‌آوری کرده و از طریق پروتکل MQTT بر بستر GPRS به سرور ارسال می‌کند. این سیستم برای انواع سیستم‌های متحرک، از خودروهای سواری تا وسایل نقلیه صنعتی، طراحی شده و با چالش‌های شبکه‌ای ایران، مانند تأخیر متغیر و قطعی‌های مکرر، سازگار است.

راه‌حل توسعه‌یافته بر چهار محور اصلی استوار است:

جمع‌آوری داده: در این ابزار با استفاده از سنسورهای ساده و ارزان قیمت وضعیت موقعیت مکانی و

وضعیت دینامیکی خودرو، شامل شتاب، جهت‌گیری و سرعت چرخش خودرو در سه محور جمع‌آوری می‌شود.

پردازش داده: در کنار داده‌های جمع‌آوری شده، سرعت خودرو نیز با کمک اطلاعات بدست آمده از

سنسور GPS و شتاب خودرو تخمین زده می‌شود. با کمک این اطلاعات، در زمان در دسترس نبودن GPS برای مدتی کوتاه، میتوان موقعیت تقریبی وسیله نقلیه را تخمین زد. برای افزایش دقت اطلاعات ژيروسکوپ و جهت گیری از فیلتر مدویک و برای افزایش دقت شتاب از میانگین متحرک نمایی استفاده می‌شود.

ارتباطات مقاوم: برای اطمینان از ارسال و دریافت مطمئن داده ها از پروتکل MQTT و تنظیم نرخ ارسال پویا با کمک وضعیت اتصال و بازخورد دریافتی استفاده میشود. این سیستم با استفاده از مکانیزمهای پشتیبان، مانند ارسال پیامک برای دادههای حیاتی در صورت قطعی شبکه، پایداری را در شرایط ناپایدار تضمین می کند.

نمایش و پایش زنده: طراحی داشبوردهای تعاملی در محیط Kibana برای نمایش بلادرنگ موقعیت خودرو، مسیر طی شده، سرعت، شتاب، جهت حرکت و سایر پارامترها (مانند شدت سیگنال)، که تحلیل و تصمیم گیری را برای کاربران تسهیل می کند.

این پروژه سیستمی نهفته برای پایش بلادرنگ موقعیت، سرعت، شتاب و جهت گیری خودروها با استفاده از سخت افزارهای مقرون به صرفه و پروتکل MQTT بر بستر شبکه GPRS توسعه داده است. محدوده پروژه شامل جمع آوری دادههای دینامیکی با سنسورهای GPS و اینرسی، پردازش آنها با الگوریتمهای سبک مانند فیلتر مدویک، انتقال پایدار دادهها در شبکههای ناپایدار ایران، و نمایش اطلاعات در داشبوردهای Kibana است. این سیستم برای کاربردهای مدیریت ناوگان، ردیابی وسایل نقلیه و لجستیک، به ویژه در مناطق با پوشش شبکه ضعیف، طراحی شده و برای وسایل نقلیه سواری تا صنعتی قابل استفاده است.

در ادامه این گزارش، ابتدا پلتفرم و ابزارهای استفاده شده معرفی می شوند (بخش ۲) و سپس تغییرات اعمال شده نسبت به فاز پرپوزال، شامل چالشها، دلایل و تجارب ناموفق، بررسی می گردند (بخش ۳). در بخش ۴، نمونههای مشابه تحلیل شده و در بخش ۵، مبانی فنی پروژه، شامل راه حلها و اثبات کارایی، شرح داده می شود. جزئیات پیاده سازی، شامل تقسیم وظایف، محیط توسعه و تغییرات ابزارها، در بخش ۶ ارائه می گردد. بخش ۷ به آزمونها، ارزیابی عملکرد و مقایسه با نمونههای مشابه اختصاص دارد. در نهایت، پیوستهای فنی (بخش ۸) و مراجع (بخش ۹) اطلاعات تکمیلی و منابع مورد استفاده را در بر می گیرند.

۲- معرفی پلتفرم و ابزارهای استفاده شده

در این پروژه، از مجموعه ای از سخت افزارها و نرم افزارهای مقرون به صرفه و در دسترس برای توسعه سیستم پایش موقعیت و وضعیت دینامیکی خودروها استفاده شده است. سخت افزار مورد استفاده به شرح زیر است:

- **سنسور MPU۹۲۵۰:** این سنسور برای جمع آوری دادههای اینرسی، شامل شتاب و جهت گیری، استفاده شده است. دقت بالا و قیمت مناسب، این سنسور را به گزینه ای مطلوب برای پروژه تبدیل کرده است.
- **ماژول SIM۸۰۸:** این ماژول برای دریافت دادههای GPS و برقراری ارتباط از طریق شبکه GPRS به کار رفته است.

- **میکروکنترلر Arduino Mega:** این میکروکنترلر با حافظه ۲۵۶ کیلوبایتی برای پردازش و مدیریت داده‌ها انتخاب شده است. حافظه چهار برابری نسبت به Arduino Uno، اجرای کدهای پیچیده‌تر را ممکن می‌سازد. همچنین زیرساخت‌های نرم افزاری استفاده شده در پروژه به شرح زیر است:
- **پروتکل MQTT:** برای ارتباط با سرور از پروتکل MQTT استفاده شده است. این پروتکل به دلیل سربار کم و کارایی بالا در شبکه‌های کم‌پهنای گزینه‌ای مناسب برای انتقال داده‌ها در بستر GPRS است.
- **بروکر HiveMQ:** از بروکر عمومی HiveMQ برای مدیریت پیام‌های MQTT بهره گرفته شده است. این بروکر به دلیل رایگان بودن و دسترسی‌پذیری مطلوب انتخاب شده است.
- **سرور پایش:** شامل یک script به زبان python که با مشترک شدن در تاپیک مربوطه، اقدام به دریافت و ذخیره سازی پیام‌ها در پایگاه داده Elasticsearch میکند و در نهایت اطلاعات دریافتی به شکل یک داشبورد تعاملی در Kibana نمایش داده میشوند. علت انتخاب این پایگاه داده پشتیبانی مناسب از داده‌های جغرافیایی و توانایی مدیریت حجم بالای اطلاعات، سازگاری آن با Kibana و همچنین آشنایی فنی تیم توسعه با این ابزارها میباشد.

۳- تغییرات نسبت به فاز پروپوزال

در فرآیند توسعه این پروژه، چالش‌های فنی و عملیاتی متعددی شناسایی شد که منجر به بازنگری تصمیمات اولیه مندرج در پروپوزال گردید. پس از تکمیل کدنویسی دستگاه Edge و کامپایل نهایی، مشخص شد که برنامه به ۳ کیلوبایت RAM و ۷۳ کیلوبایت حافظه نیاز دارد، در حالی که برد Arduino Uno پیشنهادی در پروپوزال تنها ۲ کیلوبایت RAM و ۳۲ کیلوبایت حافظه ارائه می‌داد. این محدودیت، اجرای کد را غیرممکن کرد و بهینه‌سازی‌های جزئی نیز کافی نبود. در نتیجه، برد Arduino Mega با ۸ کیلوبایت RAM و ۲۵۶ کیلوبایت حافظه، که از نظر پردازنده مشابه Uno بود، جایگزین شد تا نیازهای پروژه برآورده شود. همچنین، سنسور MPU۶۰۵۰ پیشنهادی فاقد قطب‌نمای مغناطیسی بود و امکان حذف شتاب جاذبه یا محاسبه دقیق جهت‌گیری خودرو نسبت به زمین را فراهم نمی‌کرد. این نقص، ما را به استفاده از سنسور MPU۹۲۵۰ سوق داد که با داشتن قطب‌نما، داده‌های دقیق‌تری برای تخمین وضعیت خودرو ارائه می‌کرد.

از سوی دیگر، راه‌اندازی بروکر اختصاصی MQTT که در پروپوزال پیشنهاد شده بود، با پیچیدگی و هزینه‌های اضافی همراه بود. پس از بررسی، تصمیم گرفتیم از بروکر عمومی و رایگان HiveMQ استفاده کنیم که با دسترسی‌پذیری بالا و بدون کاهش کیفیت، هزینه‌ها و پیچیدگی پیاده‌سازی را کاهش داد. علاوه بر این، بازه‌های

زمانی ارسال داده در پرپوزال (چند ثانیه) به دلیل تأخیر ذاتی شبکه GPRS (تا ۱ ثانیه در بدترین حالت) بی فایده بود. بنابراین، بازه‌های زمانی به ۳۰ ثانیه در بهترین حالت شبکه تا ۵ دقیقه در بدترین حالت تنظیم شد تا با شرایط واقعی شبکه سازگار باشد. تلاش‌های اولیه برای استفاده از برد Uno و سنسور MPU۶۰۵۰ و همچنین بازه‌های زمانی کوتاه به دلیل محدودیت‌های ذکر شده ناموفق بود، اما این تجارب به انتخاب راه‌حل‌های مناسب‌تر و بهبود کارایی سیستم منجر شد.

4- نزدیک‌ترین نمونه‌های مشابه

در سال‌های اخیر، نمونه‌های متنوعی از سامانه‌های پایش و ردیابی وسایل نقلیه مبتنی بر فناوری GPS و شبکه‌های سلولی توسط افراد مختلف توسعه داده شده‌اند که از جمله آنها میتوان به سامانه [1] GPS Tracer اشاره کرد که به کمک یک سنسور GPS امکان پایش موقعیت خودرو، ارسال آنها به کمک شبکه GPRS و نمایش آنها توسط یک سامانه تحت وب را دارد. همچنین ابزار دیگری [2] توسعه داده شده که علاوه بر پایش موقعیت خودرو بر بستر اینترنت امکان ارسال موقعیت به صورت sms را نیز دارا است.

ابزار توسعه داده شده علاوه بر فراهم کردن امکان پایش موقعیت خودرو، امکان پایش وضعیت دینامیکی خودرو را نیز فراهم میکند. همچنین در صورت در دسترس نبودن GPS به دلیل وجود نویز یا موانع محیطی، این ابزار امکان تخمین موقعیت مکانی خودرو را داراست که در شرایط اختلال در این سامانه میتواند مفید واقع شود. همچنین تنظیم نرخ ارسال به صورت پویا و امکان استفاده از پیامک به عنوان مسیر پشتیبان قابلیت اطمینان این ابزار را افزایش میدهد.

۵- مبانی فنی پروژه

سیستم پیشنهادی برای پایش موقعیت و وضعیت دینامیکی خودروها بر پایه جمع‌آوری داده‌های دقیق از سنسورهای مقرون به صرفه استوار است، که شامل داده‌های مکانی از GPS و اطلاعات اینرسی از سنسورهای IMU می‌شود. این راه‌حل‌ها با تمرکز بر دقت، پایداری و هزینه پایین طراحی شده‌اند تا در شرایط شبکه‌ای چالش‌برانگیز ایران کارآمد باشند.

داده‌های GPS از طریق ماژول SIM808، که یک ماژول ترکیبی GSM/GPRS و GPS است، برداشت می‌شوند. این ماژول با پشتیبانی از چهار باند فرکانسی (۸۵۰/۹۰۰/۱۸۰۰/۱۹۰۰ مگاهرتز) برای ارتباطات GPRS و ۲۲ کانال ردیابی / ۶۶ کانال کسب سیگنال برای GPS، امکان دریافت موقعیت مکانی دقیق را فراهم می‌کند.

دقت افقی این ماژول کمتر از ۲.۵ متر است، با دقت سرعت ۰.۰۵ متر بر ثانیه (با استفاده از DGPS) و دقت زمانی ۱۰ نانوثانیه. نرخ نمونه‌برداری GPS (sampling rate) در این ماژول ۱ هرتز است، که برای کاربردهای ردیابی خودرو مناسب است و امکان به‌روزرسانی موقعیت در فواصل منظم را بدون مصرف بیش از حد انرژی فراهم می‌کند. این ماژول همچنین از A-GPS برای موقعیت‌یابی در محیط‌های داخلی یا با سیگنال ضعیف پشتیبانی می‌کند، که حساسیت ردیابی آن را به -۱۶۵ دسی‌بل میلی‌وات می‌رساند. داده‌های خروجی شامل مختصات جغرافیایی (طول و عرض جغرافیایی)، سرعت و جهت حرکت است، که مستقیماً به میکروکنترلر Arduino Mega منتقل می‌شوند. سنسور MPU9250، یک واحد اندازه‌گیری اینرسی (IMU) ۹ محوری، برای ثبت داده‌های اینرسی خودرو به کار گرفته شده است. این سنسور شامل یک شتاب‌سنج ۳ محوری، یکژیروسکوپ ۳ محوری و یک مغناطیس‌سنج ۳ محوری است، که در یک بسته‌بندی سیستم در پکیج (SiP) ادغام شده‌اند. مشخصات فنی شتاب‌سنج شامل محدوده‌های قابل برنامه‌ریزی تا ± 16 g با حساسیت تا ۱۶۳۸۴ LSB/g است، که امکان اندازه‌گیری دقیق شتاب‌های خطی را فراهم می‌کند.ژیروسکوپ دارای محدوده‌های تا ± 2000 درجه بر ثانیه با حساسیت تا ۱۳۱ LSB/ (درجه بر ثانیه) است، که برای تشخیص چرخش‌های زاویه‌ای مناسب است. مغناطیس‌سنج (مگنتومتر) با حساسیت ۰.۱۵ میکروتسلا و تبدیل ۱۶ بیتی، داده‌های میدان مغناطیسی را برای تعیین جهت جغرافیایی ارائه می‌دهد. دقت کلی سنسور برای شتاب‌سنج حدود ± 0.05 g، برایژیروسکوپ ± 3 درجه بر ثانیه و برای مغناطیس‌سنج ± 5 میکروتسلا است، که این مقادیر آن را برای کاربردهای خودرو مانند تخمین موقعیت نسبی ایده‌آل می‌کند. نتایج خروجی سنسور شامل شتاب خطی در سه محور، نرخ چرخش زاویه‌ای (ژیروسکوپ) و جهت‌گیری مغناطیسی (قطب‌نما) است، که این داده‌ها برای جبران کمبودهای GPS در شرایط سیگنال ضعیف حیاتی هستند.

بعد از راه اندازی ماژول، ابتدا کالیبراسیون MPU9250 برای حذف خطاهای سیستماتیک انجام می‌شود. در این فرآیند، سنسور برای یک بازه زمانی کوتاه 30 ثانیه ای در حالت سکون قرار می‌گیرد و میانگین خروجی‌های شتاب‌سنج وژیروسکوپ به عنوان افست (offset) ذخیره می‌شود. این افست‌ها سپس از داده‌های خام کم می‌شوند تا اثرات نویز ذاتی سنسور و تغییرات محیطی (مانند دما یا ارتعاشات) جبران شوند. برای کاهش نوسانات ناگهانی در داده‌های شتاب وژیروسکوپ، فیلتر میانگین متحرک نمایی اعمال می‌شود و با وزندهی بیشتر به داده‌های اخیر تغییرات ناگهانی ناشی از نویز را نرم می‌کند، بدون اینکه تأخیر قابل توجهی ایجاد کند.

در مرحله بعد فیلتر Madgwick اعمال می‌شود که یک الگوریتم همجوشی سنسور (sensor fusion) است که داده‌های شتاب‌سنج،ژیروسکوپ و مغناطیس‌سنج را ترکیب می‌کند تا تخمینی دقیق از جهت‌گیری ارائه دهد.

این فیلتر بر پایه گرادیان نزولی عمل می‌کند و با حل معادلات کواترنیون (quaternion)، خطاهای انباشته ژيروسکوپ را با داده‌های شتاب و میدان مغناطیسی جبران می‌کند.

شتاب اندازه‌گیری شده توسط MPU9250 در فریم محلی (محورهای سنسور) است، که برای کاربردهای جهانی (مانند تخمین موقعیت نسبی) باید به فریم جهانی (نسبت به زمین) تبدیل شود. این تبدیل با استفاده از ماتریس چرخش کواترنیون انجام می‌شود، که شتاب محلی را به مختصات جهانی می‌برد. دلیل این کار، همخوانی داده‌ها با سیستم‌های ناوبری جهانی مانند GPS است.

شتاب جاذبه (۹.۸۱ متر بر مجذور ثانیه) همیشه در داده‌های شتاب‌سنج وجود دارد و می‌تواند اندازه‌گیری‌های واقعی حرکت خودرو را تحریف کند. نیاز به حذف آن برای محاسبه دقیق شتاب خالص ناشی از حرکت است، به‌ویژه در تخمین سرعت و موقعیت. این کار با کسر بردار جاذبه (حاصل از جهت‌گیری سنسور) از داده‌های خام انجام می‌شود.

در ادامه از Dead Reckoning برای تخمین موقعیت و سرعت در غیاب GPS استفاده می‌شود، که سرعت با انتگرال‌گیری از شتاب (پس از حذف جاذبه) و جابجایی با انتگرال‌گیری دوگانه محاسبه می‌شود. این روش می‌تواند به شرط ثابت ماندن وضعیت دینامیکی خودرو، در بازه‌های کوتاه در غیاب GPS به دلیل وجود نویز محیطی یا ضعیف بودن سیگنال تخمین خوبی از موقعیت خودرو ارائه دهد.

برای جلوگیری از خطای انباشته در شرایط ایستا، مکانیزمی برای تشخیص سکون خودرو پیاده‌سازی شده است. اگر شتاب و سرعت اندازه‌گیری شده برای یک بازه طولانی (3 ثانیه) نزدیک به صفر باشد (کمتر از ۱ متر بر مجذور ثانیه برای شتاب و ۰.۵ متر بر ثانیه برای سرعت، با توجه به نویز حرکت‌های کوتاه)، سرعت تخمینی به صفر تنظیم می‌شود. این مکانیزم خطای ناشی از نویز سنسور را در حالت سکون حذف می‌کند.

پس از جمع‌آوری و پردازش داده‌ها، سیستم به مرحله ارسال اطلاعات می‌رسد که بر پایه پروتکل MQTT و شبکه GPRS بنا شده است. این مرحله با هدف انتقال داده‌های پویای خودرو به سرور مرکزی طراحی شده و مواردی مانند زمان، شتاب و سرعت در چارچوب جهانی، نرخ چرخش زاویه‌ای، جهت‌گیری (با زوایای اولیه)، موقعیت مکانی، وضعیت تخمینی موقعیت، تازگی موقعیت، قدرت سیگنال (از ۰ تا ۱۰۰) و وضعیت باتری (درصد ۰ تا ۱۰۰ یا مقدار منفی برای نشان‌دهنده منبع تغذیه آداپتور) را پوشش می‌دهد. قدرت سیگنال و وضعیت باتری مستقیماً از ماژول SIM808 خوانده می‌شود، در حالی که زمان نیز از همین ماژول گرفته می‌شود. سرعت همان‌طور که پیشتر توضیح داده شد تخمین زده می‌شود و فیلدهای موقعیت نشان می‌دهند که آیا داده‌ها مستقیماً از GPS هستند یا تخمینی. این ساختار امکان ارسال کارآمد و مقاوم داده‌ها در برابر ناپایداری‌های شبکه را فراهم می‌کند.

پروتکل MQTT (Message Queuing Telemetry Transport) یک پروتکل سبک و مؤثر برای ارتباطات دستگاه به دستگاه در شبکه‌های با پهنای باند کم است که توسط OASIS استاندارد شده و برای کاربردهای اینترنت اشیا توسعه یافته. این پروتکل بر مدل انتشار/اشتراک استوار است و با سربار پایین (حداقل ۲ بایت برای سرآیند)، انتقال داده‌های کوچک را با تأخیر نسبی کم ممکن می‌سازد، که آن را برای سیستم‌های پایش خودرو بر بستر GPRS مناسب می‌کند. MQTT با سطوح مختلف کیفیت خدمات، پایداری را در شرایط شبکه‌ای دشوار افزایش می‌دهد و مصرف انرژی را کاهش می‌دهد.

MQTT شامل سه بخش اصلی است: انتشاردهنده (Publisher)، مشترک (Subscriber) و کارگزار (Broker). انتشاردهنده داده‌ها را به موضوعات خاص ارسال می‌کند، مشترک برای دریافت آن‌ها از موضوعات ثبت‌نام می‌کند و کارگزار به عنوان واسطه، پیام‌ها را بر اساس موضوعات توزیع می‌کند. در سیستم ما، دستگاه لبه (میکروکنترلر Arduino Mega) نقش انتشاردهنده را ایفا می‌کند و داده‌های وضعیت خودرو را به موضوع مربوطه منتشر می‌نماید. برای کارگزار، از بروکر عمومی HiveMQ بهره گرفته‌ایم که یک بروکر رایگان با ویژگی‌های برجسته‌ای مانند دسترسی پذیری بالا (uptime نزدیک به ۹۹.۹۹٪)، مقیاس‌پذیری افقی برای مدیریت میلیون‌ها اتصال، امنیت داخلی با پشتیبانی از TLS و احراز هویت است و برای پروژه‌های اینترنت اشیا بدون هزینه اضافی ایده‌آل است. مشترک نیز سرور ما است که در ادامه توصیف خواهد شد و داده‌ها را برای ذخیره و نمایش دریافت می‌کند.

در پروتکل MQTT، سطوح کیفیت خدمات (QoS) برای مدیریت تحویل پیام تعریف شده‌اند: QoS ۰ (حداکثر یک بار بدون تأیید)، QoS ۱ (حداقل یک بار با تأیید) و QoS ۲ (دقیقاً یک بار با تأیید دوجانبه). در پروژه ما، QoS ۱ انتخاب شده تا تعادلی میان کارایی و اطمینان ایجاد کند. با این حال، ارسال مجدد (retransmission) برای کار ما مناسب نیست، زیرا تأخیر شبکه GPRS (تا ۱ ثانیه یا بیشتر) می‌تواند زمان ارسال مجدد را طولانی کند و در این فاصله، داده‌های تازه تولید شود که می‌تواند جایگزین مناسبی باشد و از انباشت پیام‌های قدیمی جلوگیری کند. سیستم با دریافت بازخورد (تأیید دریافت یا ACK از بروکر) از موفقیت یا عدم موفقیت ارسال آگاه می‌شود؛ در صورت خطا، از پیامک (SMS) به عنوان روش جایگزین برای انتقال داده‌های کلیدی استفاده می‌کند تا اطلاعات از دست نرود. علاوه بر این، در صورت سه خطای پیاپی، سیستم فاصله ارسال پیام را از مقادیر پیش‌فرض ۳۰ ثانیه، ۲.۵ دقیقه و ۵ دقیقه تغییر می‌دهد تا به پایدارترین حالت برسد. در بدترین شرایط (۵ دقیقه)، همیشه پیامک نیز ارسال می‌شود تا اطمینان کامل حاصل گردد. در مقابل، پس از ۲۰ تلاش موفق پیاپی، سیستم فاصله ارسال را کوتاه می‌کند تا با بهبود شرایط سیگنال، دستگاه با حداکثر بهره‌وری عمل کند.

در سمت سرور، یک اسکریپت پایتون نوشته شده که به موضوع مربوطه در بروکر HiveMQ مشترک می‌شود، داده‌ها را دریافت می‌کند و آن‌ها را در پایگاه داده Elasticsearch ذخیره می‌نماید. Elasticsearch یک موتور جستجو و تحلیل توزیع شده بر پایه Lucene است که برای ذخیره و ایندکس گذاری داده‌های بزرگ طراحی شده و مقیاس پذیری بالایی دارد، در حالی که Kibana رابط کاربری بصری برای Elasticsearch است که امکان ایجاد داشبوردهای تعاملی و تحلیل داده‌ها را فراهم می‌کند. این ترکیب (معروف به ELK Stack) با قابلیت‌های جستجوی سریع، تجمیع داده‌های زمانی و پشتیبانی از داده‌های مکانی، برای ذخیره و دسترسی به اطلاعات پویای خودرو بسیار مناسب است و بدون نیاز به زیرساخت پیچیده، کارایی بالایی ارائه می‌دهد.

در نهایت، یک داشبورد Kibana وضعیت خودرو را به‌طور تعاملی و با تأخیر کم به کاربر نشان می‌دهد، که مزایایی مانند رابط کاربرپسند برای تحلیل بلادرنگ و سفارشی‌سازی آسان ویجت‌ها برای نمایش موقعیت، سرعت و شتاب را به همراه دارد.

۶- جزئیات پیاده‌سازی

تیم پروژه متشکل از سه عضو، محمدرضا بخشایش، امین باهنر و علی پرویزی، بود که وظایف بر اساس تخصص و توافق تیمی تقسیم شد. محمدرضا بخشایش مسئولیت خواندن و پردازش اطلاعات سنسورها را بر عهده داشت و همچنین در نوشتن سیستم ارتباطی Edge با علی پرویزی همکاری کرد. امین باهنر نوشتن اسکریپت Subscriber، استقرار ELK Stack و ساخت داشبورد Kibana را انجام داد. علی پرویزی علاوه بر همکاری در سیستم ارتباطی، مسئولیت نوشتن گزارش نهایی را به عهده گرفت.

برای کدنویسی، از PlatformIO IDE بهره گرفته شد که یک محیط توسعه یکپارچه بر پایه Visual Studio Code است و ویژگی‌هایی مانند تکمیل کد هوشمند، کامپایل کراس پلتفرم بدون وابستگی خارجی، ابزارهای دیباگ پیشرفته (مانند کاوشگر متغیرها و نقاط توقف شرطی) و پشتیبانی از میکروکنترلرهای مختلف را فراهم می‌کند. این IDE رایگان و منبع باز است و برای پروژه‌های ++embedded C/C++ ایده آل می‌باشد. تست‌ها در محیط آزمایشگاهی با استفاده از آداپتور برای تأمین برق برد ها انجام شد، اما سیستم قابلیت کار با باتری لیتیومی ۳۰۰۰ میلی آمپر ساعتی تا ۴ ساعت را نیز دارد.

سنسور MPU9250 با پروتکل I2C به میکروکنترلر متصل می‌شود؛ پایه‌های SDA و SCL سنسور به پایه‌های A20 و A21 برد Mega (که برای I2C طراحی شده‌اند) وصل می‌گردند. تغذیه سنسور از پین ۳.۳ ولتی برد Mega تأمین می‌شود تا ولتاژ مناسب فراهم گردد. ارتباط با ماژول SIM808 از طریق پورت سریال UART

انجام می‌گیرد؛ پایه TX ماژول به پین ۱۳ و RX به پین ۱۲ برد Mega متصل می‌شود (این پین‌ها در کد برنامه قابل تغییر هستند). برای جلوگیری از نویز، پایه GND دو برد نیز به یکدیگر وصل می‌شود.

در بخش نرم‌افزاری، با استفاده از PlatformIO، کدهای ++C برای خواندن داده‌های سنسورها (مانند شتاب و جهت‌گیری از MPU9250 و موقعیت از SIM808) نوشته شد. جریان داده‌ها به این صورت است: داده‌های خام از سنسورها به میکروکنترلر منتقل می‌شود، پردازش می‌گردد (مانند اعمال فیلتر مدویک)، سپس از طریق MQTT به بروکر HiveMQ ارسال می‌شود. اسکریپت پایتون در سرور داده‌ها را دریافت و در Elasticsearch ذخیره می‌کند، و در نهایت Kibana برای نمایش استفاده می‌شود. صحت‌سنجی با تست‌های واحد (مانند مقایسه داده‌های خروجی با مقادیر مرجع) و لاگ‌های سریال انجام شد، که امکان بازتولید سیستم را فراهم می‌کند.

در برنامه‌نویسی، از کتابخانه‌های زیر استفاده شد: knolleary/PubSubClient برای مدیریت اتصال MQTT و انتشار/اشتراک پیام‌ها در Arduino، که امکان ارتباط سبک و کارآمد با بروکر را فراهم می‌کند؛ vshymansky/TinyGSM برای مدیریت ارتباطات GSM/GPRS در ماژول SIM808، که شامل تنظیمات شبکه و ارسال داده‌ها می‌شود؛ و hideakitai/MPU9250 برای خواندن و پردازش داده‌های سنسور MPU9250، که توابع آماده برای کالیبراسیون و استخراج شتاب،ژیروسکوپ و مغناطیس‌سنج ارائه می‌دهد. با این حال، برای خواندن GPS از SIM808، هیچ کتابخانه‌ای پیدا نکردیم که بتواند بدون اختلال همزمان دو وظیفه خواندن GPS و ارسال/دریافت اطلاعات را انجام دهد؛ بنابراین، این بخش را به‌صورت دستی با ارسال و پارس کردن دستورات AT پیاده‌سازی کردیم، که شامل دستوراتی مانند AT+CGPSPWR برای روشن کردن GPS و AT+CGPSINF برای دریافت داده‌های موقعیت بود.

۷- آزمون، ارزیابی و مقایسه عملکرد

به‌منظور ارزیابی عملکرد پروژه، بر جنبه‌های کلیدی سیستم شامل دقت موقعیت‌یابی در حالت Dead Reckoning، تأخیر در ارسال داده‌ها و کاربرپسندی داشبورد Kibana تمرکز شد. آزمون‌ها در دو حالت شرایط واقعی و شبیه‌سازی‌شده اجرا گردید. شاخص‌های ارزیابی شامل خطای موقعیت، میانگین تأخیر و سهولت استفاده تعیین شدند.

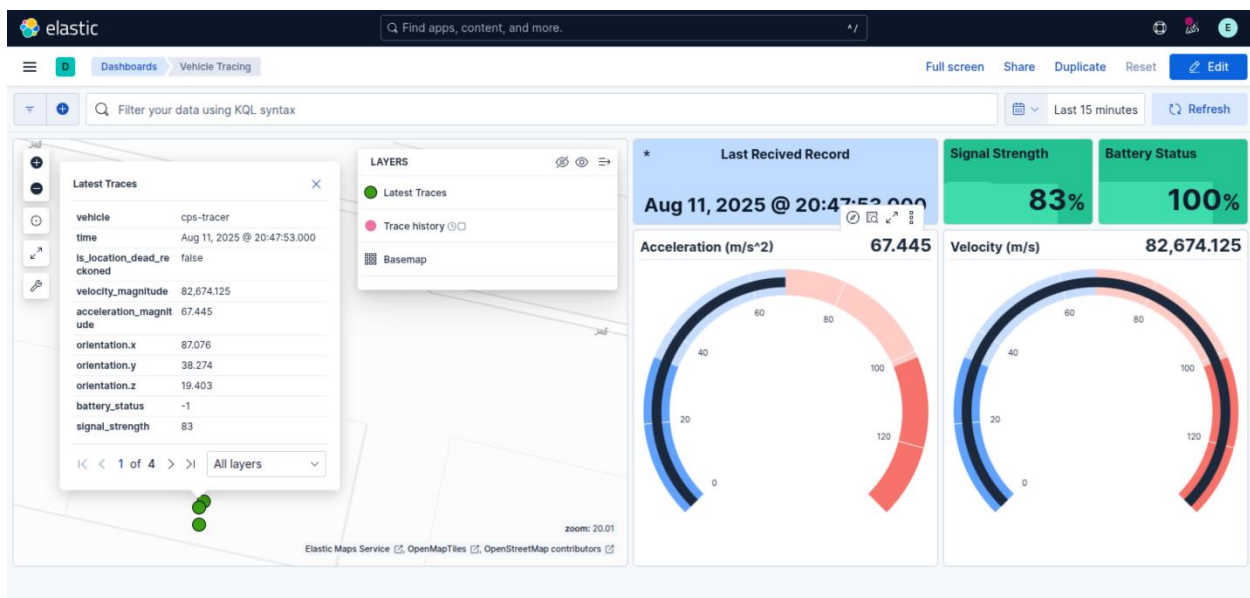
برای سنجش عملکرد Dead Reckoning، سیستم در محیط داخلی خانه فعال گردید و به‌مدت یک دقیقه با سیگنال GPS روشن جهت جمع‌آوری داده‌های اولیه کار کرد. سپس آنتن GPS جدا شد و با سرعت تقریبی یک متر بر ثانیه (معادل سرعت راه‌رفتن)، به‌مدت پنج دقیقه حرکت در محیط انجام شد. موقعیت‌ها ثبت و با داده‌های مرجع (GPS تلفن همراه) مقایسه گردید تا خطای موقعیت محاسبه شود.

همچنین برای سنجش تأخیر ارسال داده‌ها از داده‌های ثبت‌شده در آزمون قبلی استفاده شد. زمان ارسال هر بسته داده از دستگاه و زمان دریافت آن در سرور مقایسه و بر اساس 10 نمونه حاصله، میانگین تأخیر محاسبه گردید. این آزمایش در حالت سکون دستگاه نیز تکرار شد

در آزمون Dead Reckoning، خطای موقعیت از 21 متر آغاز و در بیشترین حالت به 146 متر رسید که برای جابه‌جایی‌های کوتاه‌مدت قابل قبول ارزیابی می‌شود. همچنین در حالت سکون دستگاه خطا از 19 متر شروع و در نهایت روی همان مقدار باقی ماند که نشان می‌دهد دستگاه توانایی تشخیص سکون و عدم بروزرسانی موقعیت در این حالت را دارد.

میانگین تأخیر در ارسال داده‌ها با 100 نمونه، 3144 میلی‌ثانیه به‌دست آمد که بیانگر کارایی مناسب پروتکل MQTT بر بستر شبکه GPRS است.

در نهایت شکل زیر نمایی از داشبورد پایش خودرو را نمایش می‌دهد



8- پیوست‌های فنی

تمامی کد های این پروژه از آدرس <https://github.com/Baxayesh/Cps-Vehicle-Tracing-System>

قابل دریافت است

9- مراجع

- [1] tklos, "GPS tracker," 31 march 2019. [Online]. Available: <https://github.com/tklos/track>.
- [2] A. Jekanyika, "Vehicle Tracking System (ESP8266 + GPS + GSM)," 20 january 2022. [Online]. Available: <https://github.com/jeky1950/Vehicle-Tracking-System-ESP8266-SIM800L-NEO-6M>.