



دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)
دانشکده مهندسی کامپیوتر و فناوری اطلاعات

پایان‌نامه کارشناسی
گرایش معماری سیستم‌های کامپیوتری

طراحی و پیاده‌سازی سامانه ردیابی مبتنی بر
اینترنت اشیا

نگارش
ساره سلطانی نژاد

استاد راهنما
دکتر بهادر بخشی

استاد داور
دکتر مهدی راستی

خرداد ۹۸

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

صفحه فرم ارزیابی و تصویب پایان نامه- فرم تأیید اعضاء کمیته دفاع

در این صفحه فرم دفاع یا تایید و تصویب پایان نامه موسوم به فرم کمیته دفاع- موجود در پرونده آموزشی- را قرار دهید.

نکات مهم:

- نگارش پایان نامه/رساله باید به **زبان فارسی** و بر اساس آخرین نسخه دستورالعمل و راهنمای تدوین پایان نامه های دانشگاه صنعتی امیرکبیر باشد.(دستورالعمل و راهنمای حاضر)
- رنگ جلد پایان نامه/رساله چاپی کارشناسی، کارشناسی ارشد و دکترا باید به ترتیب مشکی، طوسی و سفید رنگ باشد.
- چاپ و صحافی پایان نامه/رساله بصورت **پشت و رو(دورو)** بلامانع است و انجام آن توصیه می شود.



به نام خدا

تاریخ: خرداد ۹۸

تعهدنامه اصالت اثر

دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)

اینجانب ساره سلطانی نژاد متعهد می‌شوم که مطالب مندرج در این پایان‌نامه حاصل کار پژوهشی اینجانب تحت نظرارت و راهنمایی استادی دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مأخذ ذکر گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است. در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، ترجمه و اقتباس از این پایان‌نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مأخذ بلامانع است.

ساره سلطانی نژاد

امضا

سپاسگزاری

با سپاس فراوان از راهنمایی‌ها و زحمات استاد ارجمند، جناب آقای دکتر بخشی، که از ابتدای راه و در طی انجام این پژوهه، با رهنمودهایشان مرا در نگار این اثر یاری نمودند.

ساره سلطانی ترثا

خرداد ۹۸

چکیده

در علم فناوری اطلاعات، مفهوم اینترنت اشیا به اشیایی با هویت خاص اطلاق می‌شود که دارای شناسه منحصر به فرد بوده و توانایی انتقال داده روی شبکه، بدون نیاز به تعامل و دخالت انسان را دارند. در واقع هدف اصلی آن هوشمند سازی اشیا و فراهم آوردن بستری است که از طریق آن، اشیا قادر به ارسال و دریافت اطلاعات با یکدیگر می‌باشند.

در سال‌های اخیر فناوری اینترنت اشیا رشد چشمگیری داشته و در زمینه‌های مختلف توانسته نیازهای متعدد و پیچیده‌ای را برطرف کند. یکی از کاربردهای اینترنت اشیا در زمینه‌ی ریدیابی اشیا متحرک می‌باشد که در حوزه‌های مختلف مانند امنیت، نظارت، حمل و نقل و ... می‌تواند مورد استفاده قرار گیرد. سیستم موقعیت‌یابی و ریدیابی امکان ارائه راه حل‌هایی مطمئن برای تامین امنیت افراد و وسایل نقلیه را فراهم آورده است و همچنین تاثیر بسزایی در بهینه شدن کیفیت نظارت و مدیریت ناوگان‌های حمل و نقل، حرکت خودروها، افراد (کودکان و سالمندان) و یا هر شی متحرک دیگر دارد. در واقع سامانه ریدیابی تکنولوژی است که امکان تعیین موقعیت دقیق و ریدیابی افراد، وسایل نقلیه و یا هر جسم متحرک دیگر را با استفاده از متدهای مختلفی مانند سامانه موقعیت‌یاب جهانی فراهم آورده است.

در این پژوهه قصد داریم سامانه‌ای را پیاده‌سازی کنیم که بتوان توسط آن موقعیت دقیق، مسیر حرکت و مکان‌های پرتردد هر جسم متحرک را در هر زمان تعیین کرد. در این سیستم، هر شی مجهز به یک مژول جی‌پی‌اس است که موقعیت مکانی خود را هر دو دقیقه یکبار از ماهواره دریافت کرده و از طریق مودم جی‌اس‌ام به سرورهای نرم‌افزاری ارسال می‌کند. سرورهای نرم‌افزاری پس از دریافت اطلاعات، آن‌ها را تحلیل می‌کنند. در این قسمت پژوهه یک نرم‌افزار تحت وب توسعه داده خواهد شد تا بتواند اطلاعات ارسالی را پردازش کرده و سپس آن‌ها را در یک پایگاه داده ذخیره کند و در انتهای اطلاعات ذخیره شده را به صورت قابل نمایش برای کاربران تبدیل کند. به این ترتیب می‌توان سرعت، مسیر حرکت و مکان‌های پرتردد شی را بر روی نقشه مشاهده کرد.

واژه‌های کلیدی:

سامانه ریدیابی بی‌درنگ، سامانه موقعیت‌یاب جهانی، GSM، SIM808، اینترنت اشیا

فهرست مطالب

عنوان

صفحه

۶	۲	مفاهیم اولیه
۷	۱-۲	اینترنت اشیا
۸	۱-۱-۲	سیر تکامل
۹	۲-۱-۲	اکوسيستم
۹	۳-۱-۲	معماری فنی اینترنت اشیا
۱۰	۴-۱-۲	کاربردها
۱۱	۲-۲	سیستم ردیابی
۱۳	۳-۲	سامانه موقعیتیاب جهانی
۱۴	۱-۳-۲	تاریخچه
۱۵	۲-۳-۲	ساختار جیپیاس
۱۶	۳-۳-۲	بخش فضایی
۱۷	۴-۳-۲	کنترل زمینی
۱۷	۵-۳-۲	دقت اندازهگیری
۱۸	۶-۳-۲	کاربردها
۲۰	۳	اجزای مورد استفاده در سیستم ردیابی
۲۱	۱-۳	مقدمه
۲۱	۲-۳	طراحی و معماری سیستم
۲۳	۳-۳	اجزاء سیستم
۲۳	۱-۳-۳	اجزاء سختافزاری
۲۹	۲-۳-۳	اجزاء نرمافزاری
۳۰	۴	پیاده‌سازی سیستم ردیابی
۳۲	۱-۴	بررسی عملکرد سیستم ردیابی
۳۲	۴-۱-۴	بررسی عملکرد مدار
۳۷	۴-۲	بررسی معماری مدار

۳۷	۳-۴ برنامه کاربردی
۳۸	۱-۳-۴ پایگاه داده
۳۸	۲-۳-۴ نمایش اطلاعات
۴۲	۴-۴ تحلیل اطلاعات
۴۳	۵ جمع‌بندی و کارهای آتی
۴۴	۱-۵ جمع‌بندی
۴۴	۲-۵ کارهای آتی
۴۶	منابع و مراجع

فهرست اشکال

صفحه

شکل

۱-۲	سیر تکامل ارتباطات موجودیت‌های جهان [۱]	۸
۲-۲	معماری اینترنت اشیا [۲]	۱۰
۱-۳	بلاک دیاگرام سیستم ردیابی [۱۶]	۲۲
۲-۳	معماری سیستم ردیابی پیشنهادی [۱۹]	۲۳
۳-۳	برد آردوبینو اونو	۲۴
۴-۳	نمایی از قسمت روبرو تراشه سیم ۸۰۸	۲۶
۵-۳	نمایی از قسمت پشت تراشه سیم ۸۰۸	۲۶
۶-۳	آنتن جی‌پی‌اس	۲۸
۷-۳	مشخصات آنتن جی‌اس‌ام [۱۶]	۲۸
۸-۳	آنتن جی‌اس‌ام	۲۸
۹-۳	نمایی از نرم‌افزار آردوبینو	۲۹
۱-۴	سیستم ردیابی طراحی شده	۳۲
۲-۴	عملکرد کد پیاده‌سازی شده بر روی آردوبینو [۱۰، ۱۱، ۱۹]	۳۳
۳-۴	فلوچارت خواندن اطلاعات GPS [۸]	۳۴
۴-۴	فلوچارت نحوه کار GSM [۸]	۳۵
۵-۴	ادامه فلوچارت نحوه کار GSM [۸]	۳۶
۶-۴	شیوه اتصال آردوبینو به ماژول سیم ۸۰۸ [۳]	۳۷
۷-۴	نمایش مسیر حرکت شی در برنامه کاربردی	۳۹
۸-۴	نمایش سرعت، زمان و تاریخ حرکت شی در هر موقعیت بر روی نقشه	۳۹
۹-۴	دریافت پیام ارسالی کاربر برای دریافت موقعیت توسط ماژول سیم ۸۰۸	۴۰
۱۰-۴	پیام دریافت شده توسط کاربر	۴۱
۱۱-۴	مشاهده موقعیت شی بر روی نقشه	۴۱
۱۲-۴	مکان‌های پرتردد شی بر روی هیت مپ	۴۲

فصل اول

مقدمه

مفهوم اینترنت اشیا^۱ به اشیایی با هویت خاص اطلاق می‌شود که دارای شناسه منحصر به فرد بوده و توانایی انتقال داده روی شبکه، بدون نیاز به تعامل و دخالت انسان را دارند. هدف اصلی آن هوشمند سازی اشیا و فراهم آوردن بستری است که از طریق آن اشیا قادر به ارسال و دریافت اطلاعات با یکدیگر می‌باشند. اینترنت اشیا به طور گستردگی به توسعه قابلیت محاسبه و ارتباطات شبکه‌ای اشیا، دستگاه‌ها، سنسورها یا هر مورد دیگری که به طور معمول به عنوان کامپیوتر در نظر گرفته نمی‌شود، اشاره دارد.

این اشیای هوشمند دارای قابلیت جمع‌آوری داده از راه دور، تحلیل و مدیریت آن‌ها هستند [۱۷].

اینترنت اشیا مجموعه وسیعی از سنسورها و عملگرهایی است که شرایط مختلف محیط را اندازه‌گیری و پردازش می‌کنند. در سال‌های اخیر فناوری اینترنت اشیا رشد چشمگیری داشته و توانسته در زمینه‌های مختلف، نیازهای متعدد و پیچیده‌ای را بطرف کند. به علت گسترش فناوری‌های جدید، تولید سنسورهای هوشمند، رشد تکنولوژی‌های ارتباطی و پیچیده شدن نیازها، اینترنت اشیا قدرت زیادی پیدا کرده و در زمینه‌های مختلف از آن استفاده می‌شود و باعث گسترش سیستم‌های هوشمند در محیط شده است [۲۱].

این سیستم‌ها برای اینکه بتوانند اثر مثبتی بر محیط بگذارند باید با یکدیگر در تعامل باشند. فناوری‌های مبتنی بر اینترنت اشیا نیازمندی‌های متفاوتی در مقایسه با سایر فناوری‌ها دارند. به طور معمول این سیستم‌ها حافظه، توان مصرفی و پهنه‌ای باند کمتری نسبت به سایر سیستم‌ها دارند. اکثر سیستم‌های هوشمند مبتنی بر باتری هستند و در مکانی دوردست قرار دارند به گونه‌ای که نمی‌توان به صورت مداوم آن‌ها را شارژ کرد. در نتیجه توان مصرفی و محدوده قابل پوشش برای این سیستم‌ها به ویژه آن‌هایی که در سطح کلان اجرا می‌شوند مانند کشاورزی هوشمند، شهر و خانه هوشمند و مسائل ردیابی، مسئله بسیار مهمی است. پروتکل‌های ارتباطی بی‌سیم متعددی وجود دارد که هر کدام ویژگی منحصر به فرد خود را دارند.

اینترنت اشیاء یک بستر ارتباطی جدید در جهت برقراری ارتباط بین اشیا هوشمند می‌باشد. معرفی این بستر موجب شده است تا امکانات جدیدی برای حل مسائلی همچون تعیین مکان و ردیابی اشیا متوجه از حمله وسایل نقلیه در سطح یک شهر، منطقه یا کشور فراهم گردد. اینترنت اشیا به سرعت در حال بدست آوردن راهکارهایی در رابطه با سناریوی ارتباط از راه دور می‌باشد و انتظار می‌رود که مبادله اطلاعات در رابطه با هر شی در شبکه‌های زنجیره‌ای منابع جهانی را آسان کند، شفافیت را افزایش دهد و کارایی‌شان را بالا ببرد.

به طور گسترده اینترنت اشیا می‌تواند به عنوان ستون اصلی سیستم‌های فرآگیر و فعال‌سازی محیط‌های هوشمند برای سادگی در تشخیص و شناسایی اشیا و بازیابی اطلاعات از اینترنت در هر زمان و در هر مکان به کار برد شود.

از یک دیدگاه مفهومی، اینترنت اشیا ممکن است با توانایی اشیا هوشمند باشد: ۱- قابلیت شناسایی (هر چیزی خود را شناسایی کند) ۲- قابلیت انتقال (هر چیزی دست به انتقال می‌زند) ۳- قابلیت تعامل (هر چیزی دست به تعامل می‌زند) یا در میان خودشان و یا با کاربران نهایی یا سایر نهادهای فعال در شبکه. اشیا معمولاً یا به صورت منحصر به فرد یا به عنوان عضوی از یک رده شناسایی می‌شوند.

یکی از مسائل مطرح امروزی، ردیابی بی‌درنگ اشیا متوجه می‌باشد که به ردیابی بی‌درنگ موقعیت فعلی یک شی متوجه معین اشاره دارد. سیستم ردیابی اشیا متوجه یک راه حل برای بسیاری از مشکلات از جمله مسائل امنیتی است. تکنولوژی است که برای مشخص کردن موقعیت شی مورد استفاده قرار می‌گیرد.

یکی از کاربردهای مهم اینترنت اشیا، سامانه‌های ردیابی است که در فناوری‌های مختلف مورد استفاده قرار می‌گیرد. سیستم‌های ردیابی برای اولین بار برای صنعت حمل و نقل به وجود آمدند. از نیازهای اساسی صاحبان این صنعت، بررسی موقعیت وسایل نقلیه است. ابتدایی‌ترین سیستم ساخته شده برای یافتن موقعیت، سیستم‌های غیر فعال بودند که اطلاعات را در حافظه‌ای ذخیره می‌کردند و دسترسی به آن‌ها تنها زمانی ممکن بود که وسیله نقلیه در دسترس باشد. این نوع سیستم‌ها برای کاربردهای بلاذرنگ مناسب نیستند چون در این کاربردها نیاز است اطلاعات بلافاصله در اختیار کاربر قرار بگیرد. برای برطرف کردن این نیاز، سیستم‌های فعال به وجود آمدند که با استفاده از یک سخت‌افزار تعییه شده در وسیله نقلیه و سرور ردیابی از راه دور این امکان را فراهم می‌کنند.

امروزه امنیت افراد و وسایل نقلیه تبدیل به یک نگرانی همگانی شده است. سیستم موقعیت‌یابی و ردیابی امکان ارائه راه حل‌هایی مطمئن برای تامین امنیت افراد و وسایل نقلیه را فراهم آورده است و همچنین تاثیر بسزایی در بهینه شدن کیفیت نظارت و مدیریت ناوگان‌های حمل و نقل، حرکت خودروها، افراد (کودکان و سالمندان) و یا هر شی متوجه دیگر دارد. در واقع سامانه ردیابی تکنولوژی است که امکان تعیین موقعیت دقیق و ردیابی افراد، وسایل نقلیه و یا هر جسم متوجه دیگر را با استفاده از متدهای مختلفی مانند سامانه موقعیت‌یاب جهانی^۲ میسر می‌سازد، البته شایان ذکر است که گاهی همین مشخص بودن موقعیت افراد و وسایل نقلیه در کنار تمام مزیت‌هایی که دارد، خطرات امنیتی را

Global Positioning System (GPS)^۳

برای آن‌ها ایجاد می‌کند [۱۹].

امنیت در سیستم حمل و نقل تنها به حمل و نقل عمومی منتهی نمی‌شود. بلکه از مهم‌ترین نگرانی‌های صاحبان وسایل نقلیه شخصی، اطمینان از امنیت وسیله نقلیه آن‌ها است. سیستم‌های ردیابی در پیشگیری از سرقت یا یافتن وسیله سرقت شده می‌توانند کمک کنند. پلیس نیز با استفاده از اطلاعاتی که سیستم ردیابی تعییه شده در وسیله نقلیه ارسال می‌کند می‌تواند موقعیت را تشخیص بدهد.

علاوه بر وسایل نقلیه، سیستم‌های ردیابی در کاربردهای نظارت از راه دور و نظارت بر محیط زیست نیز نقش مهمی دارند. به عنوان مثال ردیابی حیوانات، انسان‌ها و موقعیت‌یابی اشیا از کاربردهای این سیستم می‌باشد. در مثال نظارت بر انسان‌ها، این سیستم برای افراد سالم‌مند که دارای بیماری‌های خاص چون آلزایمر هستند و احتمال گم کردن مسیر برای آنها بالا است، یا برای امنیت کودکان می‌تواند بسیار مفید باشد. خانواده‌ها می‌توانند از این سیستم برای یافتن موقعیت سالم‌مند یا کودک خود استفاده کنند [۲۰].

سیستمی که در این پروژه پیاده‌سازی کرده‌ایم می‌تواند در موارد مختلف مورد استفاده قرار بگیرد. یکی از کاربردهایی که می‌توان برای اینترنت اشیا متصور شد، پیاده‌سازی سامانه‌ای است که بتوان توسط آن موقعیت دقیق و مسیر حرکت هر جسم متحرک را در هر زمان تعیین کرد. در این پروژه قصد داریم به ساخت یک سیستم ردیابی بپردازیم که قادر است موقعیت دقیق و مسیر حرکت یک شی متحرک را مشخص کند.

در انجام این پروژه ارتباط ما به صورت یک طرفه خواهد بود به این صورت که به طور پیوسته مختصات مکانی شی متحرک توسط مژول جی‌پی‌اس^۳ اندازه گرفته و به یک سرور فرستاده می‌شود. این مژول به طور پیوسته با ماهواره برای گرفتن مختصات مکانی در ارتباط است. داده‌های جی‌پی‌اس به آردینو فرستاده می‌شود. در نهایت مودم جی‌اس‌ام^۴ این اطلاعات را برای سرورهای نرم‌افزاری ارسال می‌کند. در این پروژه سرورهای نرم‌افزاری پس از دریافت اطلاعات، آن‌ها را تحلیل می‌کنند و درخواستی از سمت سرور نخواهیم داشت و ارتباط ما به صورت یک‌طرفه خواهد بود. در این قسمت پروژه یک نرم‌افزار تحت وب توسعه داده خواهد شد تا بتواند اطلاعات ارسالی را پردازش و در پایگاه داده ذخیره کند و در انتهای اطلاعات ذخیره شده را به صورت قابل نمایش برای کاربران تبدیل کند. در واقع برنامه کاربردی نوشته شده با استفاده از داده‌های ذخیره شده، موقعیت را در نقشه نمایش می‌دهد. از این طریق می‌توان موقعی کنونی شی یا شخص مورد نظر را پیدا کرد. در انتهای موقعیت فعلی و مسیر حرکت فرد بر روی

Global Positioning System (GPS)^۳

Global system for mobile communications (GSM)^۴

نقشه نمایش داده خواهد شد. یکی دیگر از کاربردهای این سیستم به دست آوردن مکان‌های پر تردد فرد از طریق تحلیل اطلاعات جمع‌آوری شده می‌باشد.

در ادامه‌ی این پایان‌نامه، در فصل دوم درباره مفاهیم مورد استفاده در این پروژه صحبت می‌کنیم و در فصل سوم معماری کلی سیستم را بیابی و اجزا تشکیل دهنده آن و سپس در فصل چهارم شیوه پیاده‌سازی این سیستم با استفاده از اجزا معرفی شده در فصل سوم را بیان می‌کنیم. در پایان نیز درباره نتایج پیاده‌سازی سیستم و کارهایی که در آینده مبتنی بر این پروژه می‌توان انجام داد صحبت می‌کنیم.

فصل دوم

مفاهیم اولیه

۱-۲ اینترنت اشیا

اینترنت اشیا که از آن به عنوان انقلاب صنعتی جدید یاد می‌شود، به دلیل تغییری که در شیوه زندگی، کار، سرگرمی و مسافرت مردم و ... ایجاد کرده، تعاملات بین دولتها و دنیای پیرامون‌شان را با دنیای مجازی و تکنولوژی نیز دگرگون ساخته است. ورود دستگاه اتومبیل با مجموعه‌ای از نرم‌افزارهای کاربردی جهت ایجاد تعامل بین کاربر، خانه‌ها و ساختمان‌های هوشمند، امکان پخش موسیقی تنها با ادای چند کلمه و هزاران کاربرد دیگر در مدیریت هوشمند شهر، حمل و نقل، کشاورزی، صنایع دفاعی، صنعت بیمه، صنایع مربوط به نفت، گاز و معدن، مدیریت انرژی، پایش و امنیت اماکن عمومی و خصوصی، بانک‌ها، بهداشت و درمان، هتل‌داری، مهر تاییدی بر اهمیت اینترنت اشیا است.

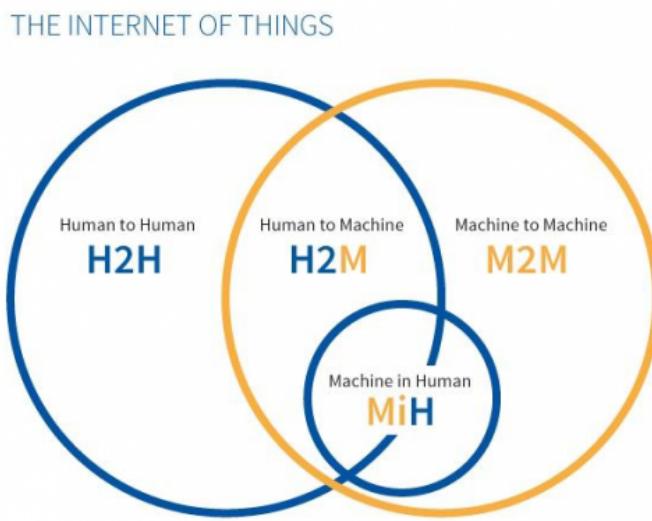
اینترنت اشیا، برای نخستین بار در سال ۱۹۹۹ توسط کوین اشتون مورد استفاده قرار گرفت و جهانی را توصیف کرد که در آن هر چیزی، از جمله اشیا بی‌جان، برای خود هویت دیجیتال داشته باشد و به کامپیوترها اجازه دهنده تا آن‌ها را سازماندهی و مدیریت کنند [۱۳].

در سال‌های بعد، تعاریف دیگری از اینترنت اشیا توسط افراد و شرکت‌های مختلف ارائه گردید. اینترنت اشیا مفهومی جدید در دنیای فناوری و ارتباطات است که به طور خلاصه می‌توان گفت، اینترنت اشیا فناوری مدرنی است که در آن برای هر موجودی (انسان، حیوان و یا اشیا) قابلیت ارسال داده از طریق شبکه‌های ارتباطی، اعم از اینترنت یا اینترانت، فراهم می‌گردد. در این فناوری، اشیا پیرامون ما قادرند از محیط اطراف خود داده‌های مفیدی را از طریق حسگرهای مختلف جمع‌آوری کرده و آن‌ها را برای پردازش و اتخاذ تصمیمات لازم به یک سیستم مرکزی منتقل کنند. در واقع ایده کلی فناوری اینترنت اشیا دریافت، ذخیره‌سازی و ارسال اطلاعات از محیط به منظور تحلیل آن‌ها و در نهایت ارائه خدمات بهتر و هوشمندتر به کاربر نهایی است. به عبارتی اینترنت اشیا را می‌توان به عنوان تکامل بعدی اینترنت دانست که جهش بزرگی در توانایی جمع‌آوری، تحلیل و توزیع داده دارد.

اینترنت اشیا یک شبکه داخلی متشکل از دستگاه‌های فیزیکی، وسایل نقلیه، ساختمان‌ها، سایر موارد الکترونیکی، نرم‌افزارها، حسگرها، محرک‌ها و یک اتصال به شبکه که اشیا را به جمع‌آوری و تبادل داده‌ها قادر می‌سازد، می‌باشد. در سال ۲۰۱۳ میلادی استانداردهای جهانی، برای اینترنت اشیا تعریف "زیرساخت‌های جامعه اطلاعاتی" را مطرح کردند. اینترنت اشیا امکان حس و کنترل شدن اشیا از راه دور را با استفاده از زیرساخت شبکه فراهم می‌سازد، فرصت ادغام مستقیم دنیای فیزیکی با سیستم‌های کامپیوتری را بالا می‌برد و در نتیجه بهبود بهره‌وری، دقت و سود اقتصادی را علاوه بر کاهش دخالت انسان به همراه دارد.

۱-۱-۲ سیر تکامل

تغییراتی که در تعامل و ارتباطات بین موجودیت‌های (اشیای) موجود در جهان، در طول زمان (قبل از ظهور اینترنت، بعد از ظهور آن تا مطرح شدن ایده اینترنت اشیا) بوجود آمده، در شکل ۱-۲ به تصویر کشیده شده است:



شکل ۱-۲: سیر تکامل ارتباطات موجودیت‌های جهان [۱]

تکامل در ارتباطات شامل مراحل زیر است:

۱. قبل از اینترنت: ارتباط انسان با انسان
۲. اینترنت محتوا: وب نسل ۱ بر روی بستر شبکه‌های IP
۳. اینترنت سرویس: وب نسل ۲ و امکان تولید محتوا توسط کاربران
۴. اینترنت افراد: شبکه‌ها و رسانه‌های اجتماعی
۵. اینترنت اشیا: اتصالات ماشین به ماشین

ارتباطات ماشین به ماشین اصطلاحی است که برای توصیف هر فناوری که دستگاه‌های شبکه را قادر به تبادل اطلاعات و انجام برخی عملیات بدون دخالت انسان می‌کند، به کار گرفته می‌شود. در واقع به عنوان بخشی از اینترنت اشیا در نظر گرفته می‌شود. پیش‌بینی‌ها نشان می‌دهد که تعداد اتصالات اینترنت اشیا در سال‌های آینده رشد قابل توجهی خواهد داشت.

مفاهیم مرتبط به اینترنت اشیا سال‌ها قبل توسط مارک ویسر در شرکت زیراکس مطرح شده بود و در قالب حوزه پردازش فراگیر در حال رشد بود. هدف حوزه پردازش فراگیر شکل‌گیری جهانی است که در آن اشیا اطراف ما (که به طور روزمره با آن‌ها سر و کار داریم) دارای قدرت پردازش بوده و به صورت بی‌سیم یا کابلی با شبکه جهانی در ارتباط باشند. دورنمای دیدگاه اولیه مارک ویسر دستیابی به سیستم‌های شبکه‌ای در حوزه فناوری اطلاعات و ارتباطات است که نهفته در محیط پیرامون، پنهان از دید کاربر و خودکار هستند. چنین سیستم‌هایی کاربر و فعالیت‌هایش را دنبال کرده و به نیازمندی‌های آن پاسخ می‌دهد. لذا به عبارتی کاربر را قادر می‌سازد با محیط فیزیکی اطرافش سازگار شود و سرویس هوشمندتری را دریافت کند.

۲-۱-۲ اکوسیستم

با توجه به تعاملی که اینترنت اشیا بین عناصر و اشیا متنوع ایجاد کرده و با درنظر گرفتن این موضوع که فناوری‌های اینترنت اشیا مختص یک صنعت و یا زنجیره تامین خاصی نیستند، این مسئله باعث ایجاد روابط پیچیده‌ای می‌شود که شامل تعداد زیادی از بازیگران خواهد بود. در این صورت بایستی با استفاده از مفهوم اکوسیستم، به این شرایط پیچیده سامان بخشید.

هدف اینترنت اشیا قابلیت بخشی به اشیا برای ارتباط با هم در همه جا، هر زمان و با هر وسیله، راه و شبکه ارتباطی است. با این وجود اینترنت اشیا، تعامل چند دنیا است که دارای اجزای مختلفی است که خود حاصل تعامل سه دنیا است و گاه‌ها از این سه دنیا تحت عنوان اجزای اینترنت اشیا نیز یاد می‌شود که عبارتند از:

۱. دنیای دیجیتال

۲. دنیای سایبری

۳. دنیای فیزیکی واقعی

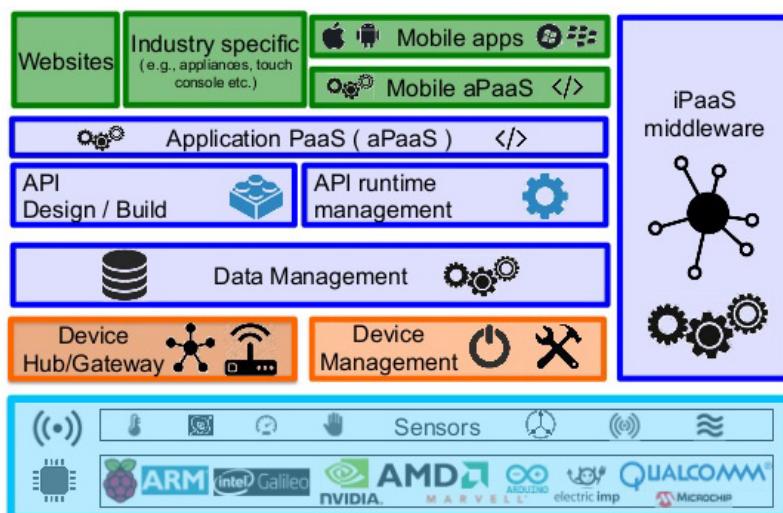
۳-۱-۲ معماری فنی اینترنت اشیا

اینترنت اشیا، ارتباطات ماشین به ماشین، وب فیزیکی و فناوری‌های متعدد دیگری که با توسعه شبکه‌های ارتباطی و تکنولوژی‌های تجمعی و تحلیل داده‌ها در سال‌های اخیر به آن‌ها پرداخته می‌شود، عملاً یک هدف ثابت را دنبال می‌کنند. این هدف بکارگیری هوشمند داده‌های تولیدی از ابزارهای متصل به شبکه

و سابقه قبلی اتصال آنها با هدف ارائه سرویس بهتر به مشتریان انتهایی است. معماری اصلی این سرویس‌ها را می‌توان به بخش‌های زیر تقسیم‌بندی کرد:

- سخت‌افزار (شامل حسگرهای عملگرها و دروازه‌های اتصال به شبکه)
- شبکه مخابراتی ارتباطی (شامل شبکه انتقال مبتنی بر دسترسی)
- بستر ابری و مدیریت سرویس (مرکز داده، پلتفرم تحلیل کلان داده و میان‌افزار ارائه سرویس)
- برنامه‌های کاربردی (شامل توسعه‌دهندگان سرویس و سرویس گیرندگان)

این معماری چند لایه‌ای در شکل زیر نشان داده شده است:



شکل ۲-۲: معماری اینترنت اشیا [۲]

۴-۱-۲ کاربردها

سرویس‌ها و کاربردهای اینترنت اشیا در حوزه‌های متنوعی مطرح هستند، که از آن جمله می‌توان به این موارد اشاره کرد: ساختمان، انرژی، خانه و مشتری، سلامت، صنعت حمل و نقل، امنیت عمومی، شبکه و فناوری اطلاعات. میزان محبوبیت کاربردهای این حوزه متفاوت است. محبوب‌ترین کاربردها به ترتیب عبارتند از: ابزارهای پوششی، شهر هوشمند، اینترنت صنعتی، توزیع هوشمند برق، سلامت، کشاورزی و دامپروری، زنجیره تامین.

۲-۲ سیستم ردیابی

همانطور که گفتیم اینترنت اشیا چهارچوبی است که در آن هر شی دارای قابلیت اتصال به اینترنت است. این اتصال به اینترنت دستگاه‌های محاسباتی تعییه شده در هر شی، آن را قادر به ارسال و دریافت اطلاعات می‌کند. اینترنت اشیا به طور گسترده به گسترش ارتباط شبکه‌ای و قابلیت محاسبه اشیا، دستگاه‌ها، حسگرها و هر چیزی که به طور عادی کامپیوتر حساب نمی‌شود، اشاره دارد. کوین اشتون، پدر اینترنت اشیا، اشاره می‌کند که "اطلاعات و داده‌ها راه عالی برای کاهش هدرروی و افزایش بهره‌وری می‌باشد و این دقیقاً آن چیزی است که اینترنت اشیا فراهم می‌کند." این اشیا هوشمند دارای حداقل نیاز انسان برای تولید، مبادله و محاسبه داده می‌باشند. زمینه‌های کاربرد تکنولوژی اینترنت اشیا روز به روز در حال افزایش است. یکی از کاربردهای اینترنت اشیا در ردیابی وسایل نقلیه است [۱۵، ۱۷].

انتظار می‌رود تعداد کل وسایل نقلیه روز به روز افزایش یابد با توجه به اینکه مالکیت آن‌ها برای افراد با توجه به اقتصاد رو به رشد کشورهایی مثل چین و هند، مقرون به صرفه می‌باشد. اما با این وجود کمبود سیستم ردیابی همچنان حس می‌شود. چنین سیستمی در زمینه‌های مختلفی از چمله امنیت خودرو شخصی، وسایل نقلیه عمومی، مدیریت ناوگان و ... کاربرد دارد [۱۸].

امروزه سیستم‌های ردیابی مختلفی در بازار وجود دارد اما آن‌ها دارای کاربرد مشخص، ناحیه کاری مشخص و عمدتاً هم بسیار هزینه‌بر می‌باشند [۱۷]. بنابراین سیستم ردیابی طراحی شده برای امنیت خودرو برای مدیریت ناوگان مناسب نمی‌باشد [۲۲]. پس باید سیستمی را پیاده‌سازی کنیم که به راحتی برای کاربردهای مختلف قابل استفاده و مقرون به صرفه باشد.

همانطور که اشاره کردیم سیستم ردیابی راه حلی برای مدیریت ناوگان و امنیت اشیا متحرک از جمله افراد، وسایل نقلیه و ... می‌باشد. این تکنولوژی برای مشخص کردن مکان شی متحرک در هر زمان از متدهای مختلفی مثل سامانه موقعیت‌یاب جهانی^۱ که به ماهواره‌های اطراف پایگاه زمین متصل می‌باشد، استفاده می‌کند. سیستم‌های ردیابی مدرن از تکنولوژی جی‌پی‌اس برای مانیتور و موقعیت‌یابی وسایل نقلیه در هر کجای زمین استفاده می‌کنند. این سیستم ردیابی در داخل خودرو یا هر شی متحرکی که می‌خواهیم آن را ردیابی کنیم، قرار می‌گیرد و امكان موقعیت‌یابی در هر لحظه را فراهم می‌آورد و اطلاعات لازم برای ردیابی را در اختیار افراد می‌گذارد. این سیستم یک دستگاه ضروری برای ردیابی اشیا متحرک مخصوصاً وسایل نقلیه است که صاحبان آن‌ها بتوانند هر زمان وسیله نقلیه خود را ردیابی کنند. هم‌چنین از این سیستم می‌توان برای ردیابی و پیدا کردن وسایل دزدیده شده استفاده نمود. داده

GPS^۱

جمع‌آوری شده از طریق این سیستم را می‌توان بر روی نقشه الکترونیکی از طریق اینترنت یا نرم‌افزارهای کاربردی مشاهده نمود [۵].

سیستم رديابي متشکل از اجزاي سخت افزاري و نرم افزاري است که با کمک آن‌ها می‌توان شى مورد نظر را رديابي کرد. اين سیستم از سه بخش اصلی تشکيل شده است: شى متحرک، بخش سخت افزاري و بخش نرم افزاري. بخش سخت افزاري متشکل از برد آردوينو، آنتن جيبي‌اس و جياس‌ام متصل به ماژول سيم ۸۰۸^۲ در داخل شى که قراره رديابي شود، قرار می‌گيرد. اين بخش سيگنال حاوی مختصات مكانی و زمانی است را به کمک آنتن جيبي‌اس از ماهواره دريافت می‌کند. مودم جياس‌ام اين اطلاعات را به سرور می‌فرستد که پس از دريافت و تحليل بر روی نقشه می‌توان مسیر حرکت را مشاهده نمود.

يکی از نگرانی‌های اصلی صاحبان وسیله نقلیه، امنیت آن‌ها می‌باشد. از اين رو همواره به دنيال سیستم امنیتی جدید و کارآمد هستند. با پیشرفت تکنولوژی امروزه قادر خواهیم بود در هر زمان وسیله نقلیه را رديابي و موقعیت آن را در هر لحظه مشاهده کنیم. سیستم رديابي با استفاده از رديابي کردن فعالیت‌های وسیله نقلیه در بازه‌های زمانی مشخص کمک بسزایی در تامین امنیت آن‌ها کرده است. اين سیستم از جيبي‌اس برای پيدا کردن اطلاعات مربوط به مكان هر شى استفاده می‌کند و سپس اين اطلاعات برای نمایش مسیر حرکت به سرور فرستاده می‌شوند.

بخش سخت افزاري سیستم رديابي در داخل وسیله نقلیه به نحوی قرار می‌گيرد که از بیرون آن قابل مشاهده نباشد. درواقع به عنوان يه بخش پنهان شده عمل می‌کند که به طور مداوم مختصات مكانی شى را به سرور می‌فرستد. در سمت سرور برنامه کاربردی تحت وب توسعه داده می‌شود که اين امكان را فراهم می‌کند مسیر حرکت شى را بر روی نقشه مشاهده کرد.

سیستم رديابي بزرگترین پیشرفت تکنولوژی در زمينه تامین امنیت می‌باشد. اين سیستم صاحبان وسال نقلیه را قادر می‌سازد بتوانند از راه دور در هر زمانی وسیله نقلیه خویش را رديابي کنند.

سیستم رديابي همچنان به عنوان دستگاه پيشگيري از سرقت و بازيابي وسیله نقلیه در ميان مردم مشهور شده است. وقتی وسیله نقلیه دزدیده می‌شود، اطلاعات ارسال شده توسط سیستم رديابي را می‌توان در اختیار پليس گذاشت تا آن‌ها به راحتی با دنيال کردن سيگنال‌های دريافتی موقعیت علی وسیله نقلیه موردنظر را پيدا کنند.

سیستم رديابي زمانی که به عنوان يك سیستم امنیتی مورد استفاده قرار می‌گيرد، می‌تواند حايگرين مناسبی برای سیستم‌های هشدار دهنده سنتی خودروها باشد. برخی از سیستم‌های رديابي قادر هستند از راه دور وسیله نقلیه را کنترل کنند که اين کنترل شامل قفل کردن در خودرو و خاموش کردن موتور

آن در موارد اضطراری می‌باشد. با توجه به پیشرفت روزافزون تکنولوژی، سیستم رديابی قادر هستند که حرکت غیر عادی وسیله نقلیه را تشخیص داده و به صاحب آن هشدار دهند.

سيستم رديابی در داخل شی که قرار است آن را رديابی کnim، قرار می‌گيرد. اين شی از طریق يک تلفن همراه که به کاربر امکان ارتباط از راه دور با سیستم تعییه شده را فراهم می‌کند. سیستم رديابی از يک سیم‌کارت برای ارسال و دریافت پیام کوتاه استفاده می‌کند. کاربر می‌تواند با استفاده از تلفن همراه خویش يک پیام کوتاه به سیم‌کارت تعییه شده در سیستم رديابی بفرستد و از طریق بتواند وسیله نقلیه خود را رديابی کند. به طور کلی عملکرد سیستم رديابی را می‌توان دو بخش کرد:

۱. رديابی وسیله نقلیه

۲. تامین امنیت وسیله نقلیه

این سیستم از يک آتن جی‌پی‌اس تشکیل شده است که رديابی واقعی شی را ممکن می‌سازد. مودم جی‌اس‌ام که مستقیم به میکروکنترلر متصل است امکان ارسال و دریافت پیام کوتاه و فرستادن داده به سرور را فراهم می‌کند. این مودم اطلاعات را دریافت می‌کند و آن را به تلفن همراه کاربر که با آن تقاضا دریافت موقعیت مکانی را کرده بود، می‌فرستد. این اطلاعات شامل طول جغرافیایی^۳، عرض جغرافیایی^۴ به همراه يک لینک می‌باشد که با کلیک کردن بر روی آن می‌توان موقعیت شی را بر روی نقشه مشاهده کرد [۱۴].

۳-۲ سامانه موقعیت‌یاب جهانی

سيستم موقعیت‌یاب جهانی توسط دولت و ارتش ایالت متحده آمریکا طراحی شد که از آن برای اهداف نظارتی استفاده می‌کردند. مأذول جی‌پی‌اس توسط وزارت دفاع ایالات متحده آمریکا و دکتر ایوان^۵ به عنوان سیستم ناوبری ماهواره‌ای اختراع شد.

Longitude^۳

Latitude^۴

Dr. Ivan^۵

سیستم موقعیت یاب جهانی منظومه‌ای متشکل از ۲۴ ماهواره است که زمین را دور می‌زند و در هر مدار ۴ ماهواره قرار دارد. راکت‌های کوچکی نیز ماهواره‌ها را در مسیر صحیح نگاه می‌دارد. این ماهواره‌ها شناسایی موقعیت جغرافیایی بین ۱۰ تا ۱۰۰ متر را امکان‌پذیر می‌سازد. این ماهواره‌ها از محاسبات ریاضی ساده‌ای برای پخش اطلاعات استفاده می‌کنند که به عنوان طول و عرض و ارتفاع جغرافیایی، توسط گیرنده‌های زمین ترجمه شده‌اند [۱۲].

سیستم جی‌پی‌اس بدون وابستگی به گیرنده‌های تلفن یا اینترنت عمل می‌کند. اگر چه با این فناوری‌ها می‌توان اطلاعات دریافتی از این سیستم موقعیت‌یابی را مناسب‌تر و کاربردی‌تر کرد. سیستم جی‌پی‌اس می‌تواند توانایی‌های حیاتی در زمینه موقعیت‌یابی برای کاربران نظامی، مدنی یا کاربران عادی در سراسر جهان فراهم کند.

پروژه جی‌پی‌اس در سال ۱۹۷۳ برای غلبه بر محدودیت‌های سیستم‌های موقعیت‌یابی پیشین شروع شد. وزارت دفاع آمریکا سیستمی را توسعه داد که به شکل پیش‌فرض ۲۴ ماهواره را به کار می‌برد. طراحی و توسعه و پشتیبانی این سیستم بر عهده وزارت دفاع ایالات متحده است.

جی‌پی‌اس در تمام شرایط به صورت ۲۴ ساعت در شب‌نه‌روز و در تمام دنیا قابل استفاده است و هیچ‌گونه بهایی بابت این خدمات اخذ نمی‌شود. ماهواره‌های جی‌پی‌اس، هر روز دو بار در یک مدار دقیق دور زمین می‌گردند و سیگنال‌های حاوی اطلاعات را به زمین می‌فرستند.

سیستم‌های مشابهی نیز وجود دارند که در حال استفاده یا طراحی هستند. سیستم روسی گلوناس مهم‌ترین آن‌ها است که تقریباً همزمان با جی‌پی‌اس تکامل یافته اما از سال ۲۰۰۸ به بهره‌برداری کامل رسیده است. اتحادیه اروپا، هند و چین نیز هر یک سیستم‌های مشابهی را در دست توسعه دارند.

۱-۳-۲ تاریخچه

از اوایل دهه ۶۰ میلادی، نیروی هوایی و نیروی دریایی آمریکا همواره در حال مطالعه یا اقدام جهت دستیابی به یک سیستم ناوبری ماهواره‌ای بوده‌اند. نیروی دریایی دو طرح عمده Timation و Transit را در دست اقدام داشته‌است. Transit توسط آزمایشگاه‌های فیزیک کاربردی جان‌هاپکینز طراحی و در سال ۱۹۶۴ به حالت عملیاتی درآمد. این سیستم اکنون اطلاعات ناوبری سطح را به صورت دو بعدی (طول و عرض جغرافیایی) و بر مبنای اصول شیفت داپلر برای کاربران دریانوردی فراهم می‌آورد. سیستم Timation طرحی تحقیقاتی با فناوری پیشرفته از آزمایشگاه تحقیقات نیروی دریایی آمریکا بود که یک سیستم ناوبری دو بعدی (طول و عرض جغرافیایی) را بر مبنای زمان‌سنجی دقیق ارائه می‌کرد. در همین

دوره زمانی، نیروی هوایی نیز تحقیقات پژوهشی را برای ارائه یک سیستم ناوبری سه بعدی (طول و عرض جغرافیایی و ارتفاع) انجام می‌داد. این سیستم بر مبنای فاصله‌یابی ماهواره‌ای به کمک «دنباله‌های دیجیتال قابل تکرار» و «نویز شبیه تصادفی» پایه‌ریزی شده بود

اولین ماهواره جی‌پی‌اس در سال ۱۹۷۸ با موفقیت به فضا پرتاب شد. هدف اصلی و اولیه از طراحی جی‌پی‌اس، اهداف نظامی بوده اما از سال ۱۹۸۰ به بعد برای استفاده‌های غیرنظامی نیز در دسترس قرار گرفت. این سیستم در سال ۱۹۹۵ و با تکمیل تعداد ماهواره‌ها به توان پیش‌بینی شده نهایی خود دست یافت. افتخار اختراع این سیستم به راجر ال استون، ایوان ای گتینگ و برادفورد پارکینسون از آزمایشگاه فیزیک پیشرفت‌ههای ایالات متحده تعلق دارد.

پیشرفت‌های تکنولوژیکی و نیازهای جدید باعث تمایل زیادی برای ارتقا و مدرنیزه کردن سیستم موجود و توسعه نسل جدید ماهواره‌ها با عنوان جی‌پی‌اس بلوک ۳ آ و نسل جدید سیستم‌های کنترل عملیاتی شده‌اند. این تغییرات از سال ۱۹۹۸ با دستور کاخ سفید شروع شدند و در سال ۲۰۰۰ با تصویب کنگره آمریکا شروع به عملیاتی شدن کردند و در نهایت به جی‌پی‌اس نسل سوم خواهند انجامید.

۲-۳-۲ ساختار جی‌پی‌اس

جی‌پی‌اس فعلی از سه بخش اساسی تشکیل شده‌است. این سه بخش اصلی عبارتند از: بخش فضایی، بخش کنترل و بخش کاربر. قسمت‌های کنترل و فضایی توسط نیروی هوایی ایالات متحده آمریکا پایه‌گذاری شده و توسعه یافته‌اند و اکنون نیز به کار خود ادامه می‌دهند [۹].

امواج منتشر شده از فضا توسط ماهواره‌های جی‌پی‌اس، توسط گیرنده‌های جی‌پی‌اس دریافت می‌شوند؛ این گیرنده‌ها به وفور در اختیار انواع کاربران قرار دارند و برای محاسبه کردن موقعیت سه بعدی (طول و عرض جغرافیایی و ارتفاع) محل مورد نظر و زمان به کار می‌روند.

• بخش فضایی: از ۲۴ تا ۳۲ ماهواره تشکیل شده است که در مدار میانی زمین قرار گرفته‌اند و همچنین شامل تأسیساتی هم می‌شود که برای آماده‌سازی و پرتاب آن‌ها به کار می‌روند.

• بخش کنترل: از یک ایستگاه اصلی کنترل زمینی، یک ایستگاه اصلی کنترل زمینی دیگر به عنوان پشتیبان، یک میزبان آنتن‌های اختصاصی و اشتراکی برای سیستم و ایستگاه‌های پایش تشکیل شده‌است.

• بخش کاربری: از صدها هزار کاربر نظامی آمریکایی و متحдан آن که از جی‌پی‌اس کدگذاری شده

برای تعیین موقعیت دقیق استفاده می‌کنند و صدها میلیون کاربر مدنی، عمومی یا علمی تشکیل شده‌است که از امکانات موقعیت‌یابی استاندارد استفاده می‌کنند.

۳-۳-۲ بخش فضایی

بخش فضایی از ماهواره‌های مستقر در مدار زمین تشکیل شده است که به اختصار ماشین‌های فضایی نیز نامیده می‌شوند. در طراحی اولیه جی‌پی‌اس ۲۴ ماشین فضایی مورد نیاز بود که در ۸ مدار دایره‌ای و در هر مدار حداکثر ۳ ماهواره قرار می‌گرفتند. بعدها این طرح تبدیل به ۶ مدار شد و در هر مدار حداکثر ۴ ماشین فضایی در نظر گرفته شد.

نقشه ۶ مداری حداکثر ۵۵ درجه انحراف مداری دارد که هر مدار 60° درجه فاصله از گره نزولی دارد. زمان مداری نصف یک روز نجومی است؛ معنی آن این است که روزانه حدود ۱۱ ساعت و ۵۸ دقیقه طول می‌کشد تا ماهواره از روی مکان قبلی یا تقریباً نزدیک آن عبور کند.

مدارها به شکلی تنظیم شده‌اند که در تمام ساعات شبانه روز و تقریباً از تمام نقاط سطح زمین، حداقل ۶ ماهواره در خط دید باشند. برای تحقق این موضوع فاصله یکسانی برای ماهواره‌های موجود در مدار مشترک در نظر گرفته نشده است. اگر ساده‌تر در نظر بگیریم فاصله زاویه‌ای بین بین ماهواره‌ها به این شکل است: 30° ، 105° ، 120° ، 105° درجه که در مجموع 360° درجه می‌شود.

ارتفاع مداری حداکثر حدود 20200 کیلومتر است، یعنی شعاع مداری حداکثر 26600 کیلومتر است. هر ماشین فضایی، در هر روز نجومی دو بار و همان مسیر قبلی را نسبت به زمین می‌پیماید. این مسئله مخصوصاً هنگام ارتقا و تکمیل سیستم خیلی کمک کننده بود چرا که حتی فقط با ۴ ماهواره و جاگیری صحیح، هر چهار ماهواره در طی چند ساعت، از یک نقطه خاص قابل رویت بودند. برای عملیات‌های نظامی، تکرار گذرهای زمینی از یک منطقه می‌تواند منجر به اطمینان از پوشش خوب منطقه نبرد باشد. در فوریه 2016 ، 32 ماهواره در سیستم جی‌پی‌اس حضور داشتند که 31 عدد از آن‌ها فعال بودند. ماهواره‌های اضافی دقت محاسبات گیرنده‌های جی‌پی‌اس برای اندازه‌گیری‌های دقیق را افزایش می‌دهند. با افزایش تعداد ماهواره‌ها چینش آن‌ها در مدارها به شکل ناهمسانی تغییر کرد. مزیت این شکل از چینش نسبت به فرم استاندارد این است که در صورت از دست رفتن یکی از ماشینها فضایی (عدم کارکرد صحیح)، در دسترس بودن سیستم کاهش نمی‌یابد و هنوز مورد اعتماد باقی می‌ماند. با وضعیت فعلی از هر نقطه زمین و در هر زمانی در حدود 9 ماهواره به شکل هم‌زنان در خط دید قرار دارند. این امر باعث افزایش قابل توجه اعتماد به دقت، نسبت به حضور حداقل 4 ماهواره، برای تعیین مکان می‌شود.

۴-۳-۲ کنترل زمینی

بخش کنترل این بخش شامل ایستگاه‌های کنترل زمینی است که دارای مختصات معلوم هستند و موقعیت آن‌ها از طریق روش‌های کلاسیک تعیین موقعیت نظیر روش تعیین فواصل بلند و روش فاصله‌سنگی ماهواره‌ای با امواج لیزر بدست آمده است. این ایستگاه‌ها وظیفه تعقیب چندجمله‌ای ریاضی به طریق کمترین مربعات، پارامترهای مداری و موقعیت ماهواره‌ها را نسبت به یک سیستم مختصات ژئودتیک ژئوسنتریک (مبداً سیستم مختصات تقریباً در مرکز زمین قرار دارد) محاسبه می‌نماید. تعداد این ایستگاه‌های زمینی ۵ عدد است که ایستگاه اصلی با نام کلرادو اسپرینگ در آمریکا قرار دارد و ۴ ایستگاه فرعی دیگر در نقاط دیگر کره زمین مستقر هستند. آخرین بخش از سیستم جی‌پی‌اس، قسمت کاربران سیستم است که خود شامل دو بخش است:

۱. آتن دریافت کننده اطلاعات ارسالی از ماهواره‌ها

۲. گیرنده (پردازش کننده اطلاعات دریافتی و تعیین کننده موقعیت محل آتن)

نرمافزار و ریزپردازنده داخل گیرنده فاصله بین آتن زمینی تا ماهواره‌های مرتبط با گیرنده را تعیین می‌کند سپس با استفاده از حداقل ۴ ماهواره موقعیت طول، عرض و ارتفاع محل استقرار آتن یا همان گیرنده تعیین می‌شود. نکته مهمی که می‌بایست مورد توجه قرار گیرد این است که ارتفاعی که جی‌پی‌اس به ما می‌دهد با ارتفاع موجود در نقشه‌ها و اطلس‌ها فرق می‌کند. ارتفاع جی‌پی‌اس نسبت به سطح مبنایی به نام بیضوی است در حالی که ارتفاع موجود در نقشه‌ها ارتفاع اورتومتریک است که از سطح دریاهای آزاد محاسبه می‌گردد. مقدار این اختلاف در بیشترین حالت در حدود ۱۰۰ متر است.

۵-۳-۲ دقیق اندازه‌گیری

دستگاه‌های جی‌پی‌اس انواع گوناگونی دارند و در وسایل مختلفی استفاده می‌شوند؛ ولی یکی از جاهایی که بیشتر در آن استفاده می‌شود گوشی‌های موبایل است که دقیق آن در شرایط مختلف متفاوت است.

به عنوان مثال:

• فضای باز: ۳ تا ۷ متر

• داخل ماشین: ۵ تا ۱۰ متر

• داخل خانه دو طبقه: ۲۰ تا ۵۰ متر

- داخل آپارتمان: ۵۰۰ تا ۱۰۰۰ متر و در شرایطی قطع
- در جنگل: ۲۰۰ تا ۱۰۰ متر
- داخل هواپیما: ۵۰ متر
- داخل تونل: قطع کامل
- بر پشت بام آپارتمان: ۴ تا ۲ متر
- زیر پل‌ها: قطع کامل

۶-۳-۲ کاربردها

اساس کار دستگاه موقعیت‌یاب تعیین موقعیت است و این کار با دریافت امواج ماهواره‌های موقعیت‌یاب که در مدارهایی به دور کره زمین در چرخش هستند، انجام می‌شود. برای تعیین موقعیت، گیرنده موقعیت‌یاب زمان‌های دریافت شده را با زمان خود مقایسه می‌کند. تفاوت این دو مشخص‌کننده فاصله گیرنده موقعیت‌یاب از ماهواره مجبور می‌باشد. این عملی است که دقیقاً یک گیرنده موقعیت‌یاب انجام می‌دهد. با استفاده از حداقل سه ماهواره یا بیشتر، موقعیت‌یاب می‌تواند طول و عرض جغرافیایی مکان خود را تعیین نماید. (که آن را تعیین دو بعدی می‌نماید) و با تبادل با چهار (و یا بیشتر) ماهواره یک موقعیت‌یاب می‌تواند موقعیت سه بعدی مکان خود را تعیین نماید که شامل طول و عرض جغرافیایی و ارتفاع می‌باشد. با انجام پشت سر هم این محاسبات، موقعیت‌یاب می‌تواند سرعت و جهت حرکت خود را نیز به دقت مشخص نماید.

امروزه در بعضی مکان‌های ایران قادر به دریافت اطلاعات تا ۱۰ ماهواره می‌باشیم و حداقل به ۴ تا ۵ ماهواره در هر زمان از شبانه روز و در هر مکان دسترسی داریم. هر قدر تعداد ماهواره‌های قابل مشاهده بیشتر شود معادلات اساسی تعیین موقعیت بیشتر خواهند شد و بنابراین زمان لازم برای تعیین موقعیت یک نقطه کاهش یافته و دقت تعیین موقعیت نیز افزایش خواهد یافت.

نکته مهمی که می‌بایست مورد توجه قرار گیرد این است که ارتفاعی که موقعیت‌یاب به ما می‌دهد با ارتفاع موجود در نقشه‌ها و اطلس‌ها فرق می‌کند. ارتفاع موقعیت‌یاب ارتفاع ژئودتیک است که نسبت به سطح مبنایی به نام بیضوی مقایسه اندازه‌گیری می‌شود، در حالی که ارتفاع موجود در نقشه‌ها ارتفاع اورتومتریک یا ارتفاع از سطح ژئوئید می‌باشد که از سطح دریاهای آزاد محاسبه می‌گردد. مقدار اختلاف

این دو مقیاس در بیشترین حالت حدود ۱۰۰ متر است. موقعیت یاب اطلاعات موقعیتی را توسط اعداد و در یک سیستم مختصاتی بیان می کند.

دستگاه جی پی اس یک رایانه کوچک است که جهت انجام امور خاصی برنامه ریزی شده است؛ بنابراین این رایانه با داشتن مختصات شما می تواند کارهای دیگری هم انجام بدهد. مثلاً می تواند زمان طلوع و غروب خورشید را در موقعیت شما بگوید. همچنین زمان طلوع و غروب ماه. شاید خیلی جالب باشد ولی جی پی اس می تواند زمان باقیمانده برای رسیدن به مقصد مورد نظر را با توجه به سرعت شما محاسبه کند. همچنین میانگین سرعت شما، بیشترین سرعت، میانگین سربالایی و سرازیری مسیر، سرعت عمودی، موقعیت منطقه از نظر شکار و ماهیگیری و شکار در هر نقطه جهان، محاسبه مساحت یک نقطه ناشناخته و برگرداندن شما از مسیر آمده را نیز می تواند انجام دهد.

فصل سوم

اجزای مورد استفاده در سیستم ردیابی

۱-۳ مقدمه

هدف اصلی پژوهش ما طراحی و پیاده‌سازی سامانه‌ای است که بتوان توسط آن موقعیت دقیق و مسیر حرکت هر جسم متحرک را در هر زمان تعیین کرد. سامانه ذکر شده باید علاوه بر عملکرد مناسب، از لحاظ هزینه هم به صرفه باشد.

برای این که بتوانیم چنین سامانه‌ای را طراحی کنیم اول باید نیازمندی‌های سامانه را تشخیص دهیم، معماری کلی سامانه موردنظر خود را به دست آوریم و سپس با استفاده از این معماری و نیازمنجی انجام شده برای پیاده‌سازی از ماثول‌های مناسب استفاده کنیم. در این فصل در قسمت ۲-۳ ابتدا طرح کلی سامانه ردیابی را توضیح می‌دهیم و سپس در بخش ۳-۳ اجزاء مورد استفاده در این طرح را معرفی می‌کنیم.

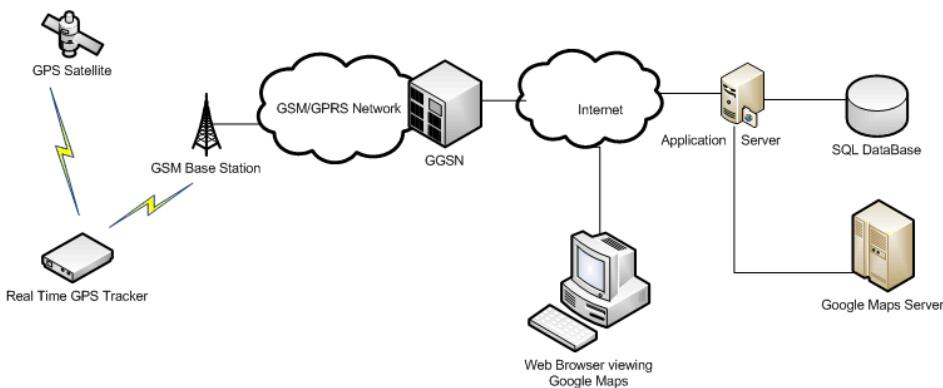
۲-۳ طراحی و معماری سیستم

در این قسمت به طراحی سیستم خود می‌پردازیم. با توجه به نیازمندی‌های پژوهش باید ماثول‌های فرستنده و گیرنده، پروتکل ارتباطی و برنامه کاربردی برای نمایش اطلاعات را مشخص کنیم. هدف اصلی یک سیستم ردیابی این است یک شی خاص را ردیابی کرده و مسیر حرکت آن را در هر زمانی بدست آوریم. در واقع سیستم ردیابی اطلاعاتی درباره مکان فعلی و سرعت شی مورد نظر را در اختیار ما می‌گذارد. در انجام این پژوهش ارتباط ما به صورت یک طرفه بوده است، به این صورت که به طور پیوسته مختصات مکانی شی متحرک اندازه گرفته می‌شود و به یک سرور فرستاده می‌شود و سپس پردازش‌های لازم در سمت سرور بر روی این اطلاعات صورت می‌گیرد. با توجه به توضیحات گفته شده می‌توان به سه قسمت اصلی در این سیستم اشاره کنیم [۶]:

- بدست آوردن موقعیت مکانی شی متحرک با استفاده از ماثول جی‌پی‌اس
- ارسال اطلاعات مکانی به سرورهای نرمافزاری توسط مودم جی‌اس‌ام
- ذخیره اطلاعات مکانی در سمت سرور و پیاده‌سازی برنامه کاربردی برای نمایش مسیر حرکت شی بر روی نقشه

همانطور که دیدیم معماری سیستم ما دارای چهار بخش اصلی است. بخش اول مربوط به گرفتن موقعیت مکانی شی از ماهواره با استفاده از ماثول جی‌پی‌اس است. بخش دوم مربوط به ارسال اطلاعات دریافتی

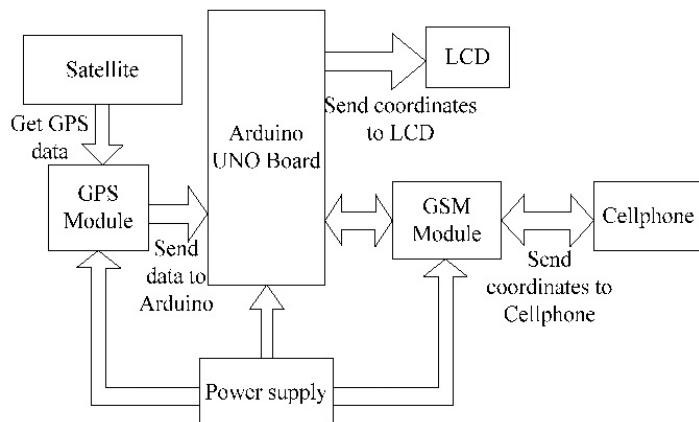
به سرور با استفاده از مودم جی‌اس‌ام است و بخش سوم هم توسعه برنامه کاربردی است که با استفاده از اطلاعات دریافت شده موقعیت شی مورد نظر بر روی نقشه نمایش داده می‌شود.



شکل ۳-۱: بلاک دیاگرام سیستم ردیابی [۱۶]

شکل ۳-۱ نمای کلی از معماری سیستم طراحی شده و ارتباط بین بخش‌های آن را نشان می‌دهد. برای انتخاب مازول‌ها لازم است وظیفه هر بخش را دقیق بدانیم و مازول مورد نظر برای آن را انتخاب کنیم [۵].

- در بخش اول لازم است ما موقعیت مکانی شی مورد نظر را به طور پیوسته اندازه بگیریم. در واقع به محض حرکت کردن شی، مازول جی‌پی‌اس به طور پیوسته اطلاعات مکانی و زمانی شی مورد نظر را از ماهواره دریافت می‌کند. سیگنال دریافتی از ماهواره ضعیف می‌باشد و لذا باید از یک آنتن برای تقویت سیگنال مورد نظر استقاده کنیم و در انتهای سیگنال تقویت شده که حاوی اطلاعات مکانی و زمانی شی متحرک می‌باشد را به برد آردوینو می‌فرستد.
- در بخش دوم اطلاعات دریافتی از مازول جی‌پی‌اس، توسط مودم جی‌اس‌ام به سمت سرور فرستاده می‌شود.
- سرورهای نرمافزاری پس از دریافت اطلاعات آن‌ها را تحلیل می‌کنند. ارتباط ما در این پروژه به صورت یک‌طرفه می‌باشد و درخواستی از سمت سرورهای نرمافزاری نخواهیم داشت. در این قسمت پروژه یک نرمافزار تحت وب توسعه داده خواهد شد تا بتواند اطلاعات ارسالی را پردازش و در پایگاه داده ذخیره کند. در قسمت آخر هم این اطلاعات ذخیره شده در صفحه وب طراحی شده نمایش داده می‌شود.



شکل ۳-۲: معماری سیستم ردیابی پیشنهادی [۱۹]

۳-۳ اجزاء سیستم

در قسمت قبل معماری سیستم را مشخص کردیم. حال اجزاء این معماری را به طور دقیق بیان و معرفی می‌کنیم.

۱-۳-۳ اجزاء سختافزاری

اجزای سختافزاری که برای پیاده‌سازی این سامانه استفاده شده است عبارتند از:

- ماژول آردوینو^۱
- ماژول سیم^۲
- آنتن جی‌پی‌اس^۳
- آنتن جی‌اس‌ام^۴

۱-۱-۳ ماژول آردوینو

آردوینو یک ریزپردازنده متن‌باز است که برای نوشتن برنامه‌هایی که با محیط و اشیا بیرون در تعامل هستند مناسب است. این برد مناسب نمونه‌سازی می‌باشد و نرم افزار و طرح سختافزار آن به صورت آزاد

SIM 808^۱

GPS Antenna^۲

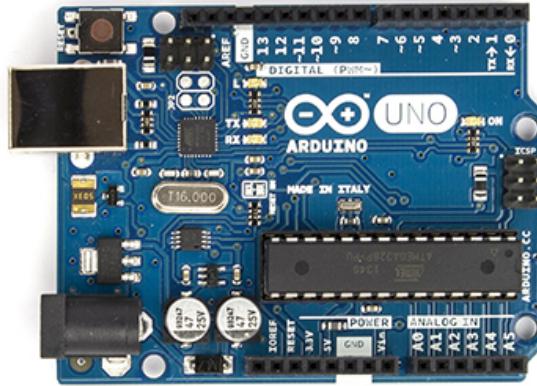
GSM Antenna^۳

در اختیار تمام افراد قرار گرفته است و هر فرد علاقه‌مند حتی با دانش و تجربه اندک در حوزه الکترونیک می‌تواند از آردوینو برای انجام پروژه‌های خود استفاده نماید.

آردوینو محیط ساده‌ای برای برنامه‌نویسی دارد که هر شخصی با اندکی آشنایی با زبان سی^۴ و سی‌پلاس‌پلاس^۵ می‌تواند در این محیط برنامه‌نویسی کند و برنامه نوشته شده را در آردوینو اجرا نماید. به میکروکنترلر آردوینو می‌توان حسگرهای مختلف متصل و آن‌ها را کنترل کرد. ریزپردازنده به کار رفته بر روی برد آردوینو بر اساس زبان برنامه‌نویسی آردوینو برنامه‌ریزی شده است و برای کدنویسی به نرم‌افزار یا کامپایلر جانبی نیازی ندارد.

آردوینو انواع مختلفی دارد که ما از آردوینو اونو^۶ در این پژوهه استفاده کرده‌ایم. R3 سومین و آخرین نسخه آردوینو اونو می‌باشد. برد آردوینو اونو یک میکروکنترلر بر پایه ATmega328 می‌باشد [۵]. ولتاژ کاری آن ۵ ولت می‌باشد. ولتاژ ورودی این برد می‌تواند در بازه ۷ تا ۲۰ ولت باشد. این برد دارای ۶ پین ورودی آنالوگ، ۱۴ پین ورودی و خروجی دیجیتال، یک پورت یواس‌بی^۷، یک ورودی منبع تغذیه و یک دکمه بازنشانی^۸ است که اجزاء اتصال بردگاهی توسعه مختلفی را فراهم می‌آورد.

در شکل ۳-۱ برد آردوینو Uno را مشاهده می‌کنید.



شکل ۳-۳: برد آردوینو اونو

C^۹
C++^{۱۰}
Arduino Uno R3^{۱۱}
USB Port^{۱۲}
Reset^{۱۳}

۲-۱-۳-۳ مژول سیم ۸۰۸

ماژول سیم ۸۰۸ یک ماژول ترکیبی از جی‌اس‌ام / جی‌پی‌آراس^۹ و ماژول جی‌پی‌اس با قابلیت پشتیبانی از چهار باند فرکانسی ۱۹۰۰/۱۸۰۰/۹۰۰/۸۵۰ مگاهرتز برای ارسال داده، پیام کوتاه و برقراری تماس صوتی می‌باشد. این ماژول دارای یک سوکت سیم‌کارت می‌باشد که سیم‌کارت در داخل آن قرار می‌گیرد. این ماژول آخرین و جدیدترین محصول شرکت SIMCOM می‌باشد که از شبکه چهار باند جی‌اس‌ام / جی‌پی‌آراس پشتیبانی و برای ردیابی ماهواره‌ای از فناوری جی‌پی‌اس استفاده می‌کند. در واقع با استفاده از مودم جی‌اس‌ام / جی‌پی‌آراس و ماژول سیم ۸۰۸ می‌توان به تبادل داده روی شبکه جی‌اس‌ام از طریق واسطه یواس‌بی پرداخت و از طریق به اطلاعات دستگاه‌های مستقر در مکان‌های دور دسترسی یافت.

طراحی فشرده این تراشه که دو سیستم مخابراتی و موقعیت‌یاب را در یک بسته ادغام می‌کند موجب کاهش هزینه و زمان برای انجام پروژه‌های مبتنی بر جی‌پی‌اس شده است. این ماژول با تکنولوژی ذخیره انرژی^{۱۰} طراحی شده است و مصرف انرژی آن در حالت خواب بسیار کم در حدود یک میلی آمپر می‌باشد. این ماژول دارای ۶۸ پین، سوکت یو اس بی، سیم‌کارت، بلوتوث می‌باشد. دارای حساسیت بالای دریافت موقعیت جهانی با ۲۲ کanal ردیابی و ۶۶ کanal گیرنده می‌باشد. علاوه بر این از A-GPS پشتیبانی می‌کند که برای موقعیت‌یابی داخل ساختمان استفاده می‌شود. این ماژول توسط فرمان AT کنترل می‌شود و از سطح منطقی ۳.۳ تا ۵ ولت پشتیبانی می‌کند [۴]. از جمله ویژگی‌های این تراشه می‌توان موارد زیر را نام برد:

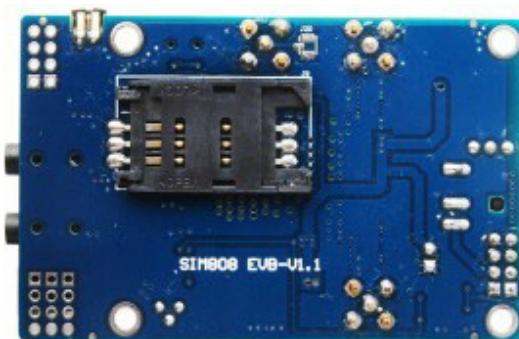
- پشتیبانی از سیم‌کارت تمامی آپراتورها
- دارای مدار کنترل شارژ
- پشتیبانی از فرکانس ساعت
- پشتیبانی از کanal‌های آنالوگ
- کم‌صرف (۱ میلی آمپر در حالت خواب)
- ولتاژ ورودی ۴.۳ تا ۴.۴ ولت
- قابلیت نصب ۳ آنتن جی‌پی‌اس، جی‌اس‌ام و بلوتوث

GSM/GPRS^۹
Power Saving^{۱۰}

در شکل ۳-۲ و ۳-۳ این تراشه را مشاهده می‌کنید.



شکل ۳-۴: نمایی از قسمت رو برو تراشه سیم ۸۰۸



شکل ۳-۵: نمایی از قسمت پشت تراشه سیم ۸۰۸

۳-۱-۳-۳ آنتن جیپیاس

بهتر است قبل از معرفی آنتن جیپیاس، شیوه موقعیت یابی توسط سیستم موقعیت یاب جهانی را به طور مختصر توضیح بدھیم. جیپیاس در واقع شامل ۲۷ ماهواره است که در اطراف زمین در حال گردش هستند که از این ۲۷ ماهواره ۳ تای آنها به صورت رزرو شده می‌باشند. هر ماهواره سیگنال‌های منحصر

به فرد و پارامترهای مداری را ارسال می‌کند و هر گیرنده‌ای که این سیگنال را دریاف کند، با رمزیشایی اطلاعات دریافتی می‌تواند موقعیت دقیق ماهواره را پیدا کند. با اتصال سیستم موقعیت‌یاب به سه ماهواره می‌توان موقعیت دو بعدی یعنی طول و عرض جغرافیایی و با اتصال به چهار ماهواره می‌توان موقعیت سه بعدی را به دست آورد [۵].

جی‌پی‌اس با دریافت سیگنال‌های ماهواره، موقعیت و مکان شی را مشخص می‌کند. برای دریافت درست سیگنال باید از آنتن استفاده شود. سیگنال‌های ماهواره‌ای جی‌پی‌اس در خطوط L1 و L2 به ترتیب دارای فرکانس‌های ۱۵۷۵.۴۲ و ۱۲۲۸ مگاهرتز می‌باشند اما قدرت سیگنال دریافتی معمولاً ضعیف بوده و در حدود ۱۶۶ دسی‌بل^{۱۱} می‌باشد که این موضوع لزوم وجود آنتن و تقویت کننده سیگنال جی‌پی‌اس را نشان می‌دهد. این آنتن سیگنال را به اندازه ۲۸ دسی‌بل تقویت می‌کند و جریان حدود ۱۰ میلی‌آمپر می‌کشد و دارای کابلی به طول ۵ متر می‌باشد که این موجب می‌شود به راحتی به هر جایی که لازم است دسترسی پیدا کند. این آنتن مغناطیسی است و می‌تواند به بالای ماشین یا هر ساختار فلزی دیگر بچسبد. دارای فرکانس کاری ۱۵۷۲.۴۲ مگاهرتز و محدوده ولتاژ ۲.۵ تا ۵.۵ ولت می‌باشد.

همانطور که گفتیم سیگنال جی‌پی‌اس بسیار ضعیف هستند و برای تقویت آن‌ها به آنتن نیاز داریم. از این رو انتخاب آنتن مناسب نقش مهمی در عملکرد جی‌پی‌اس دارد. یک واحد جی‌پی‌اس به یک دید واضح و بدون مانع با آسمان نیاز دارد تا بتواند بهترین سیگنال‌هایی که موجب می‌شود با ماهواره ارتباط برقرار کند را دریافت کند. جی‌پی‌اس برای کابل‌های طویل از مبدل بالا/پایین استفاده می‌کند. به این صورت که آنتن سیگنال جی‌پی‌اس را دریافت می‌کند، آن را به یک فرکانس پایین‌تر تبدیل می‌کند و سپس از طریق کابل آن را می‌فرستد. در سمت گیرنده جی‌پی‌اس هم یک مبدل بالا وجود دارد که فرکانس آن را به فرکانس سیگنال اصلی برمی‌گرداند و آن را به گیرنده جی‌پی‌اس می‌فرستد. در شکل ۳-۴ این آنتن را مشاهده می‌کنید.

۴-۱-۳-۳ آنتن جی‌پی‌اس

ارتباطات سیستم موقعیت‌بای جهانی وابسته به آنتن می‌باشد. آنتن به سیگنال‌های ارتباطی اجازه می‌دهد، ارسال و دریافت شوند. آنتن مورد استفاده در این پروژه در چهار باند فرکانسی با بهره ۲ دسی‌بل کار می‌کند. در واقع فرکانس کاری آن ۸۹۰، ۹۶۰، ۱۷۱۰، ۱۸۸۰ مگاهرتز می‌باشد [۵].

شکل ۳-۵ برخی از مشخصات آنتن را نشان می‌دهد. در شکل ۳-۶ این آنتن را مشاهده می‌کنید.

dB^{۱۱}



شکل ۳-۶: آنتن جی پی اس

Item	Value
Frequency	850 MHz-900 MHz-2.1 GHz-1800 MHz-1900 MHz
Impedance	50 Ohms
Mounting	on glass
Polarization	horizontal
Gain	2.14dBi
VSWR	< 2:1
Power handling	25W
Connector	RPSMA Male
Size	117mm x12.5mm x 4mm
Operating temperature	-40°C to +85°C

شکل ۳-۷: مشخصات آنتن جی اس ام [۱۶]

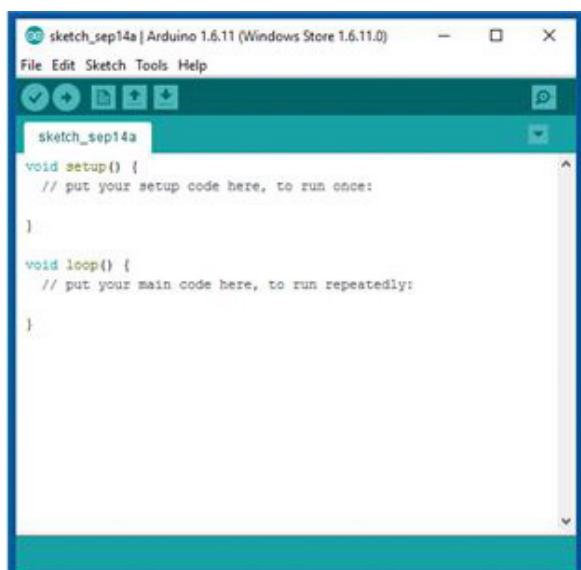


شکل ۳-۸: آنتن جی اس ام

۲-۳-۳ اجزاء نرم افزاری

۱-۲-۳-۳ نرم افزار Arduino IDE

نرم افزار مورد استفاده برای برنامه ریزی آردوینو، نرم افزار Arduino IDE می باشد که در شکل ۷-۳ نمای کلی از ظاهر این برنامه را مشاهده می کنید. با استفاده از زبانی شبیه سی میتوان برنامه مورد نیاز را نوشت و بعد از کامپایل، کد هگز تولید شده بر روی آردوینو باز می شود. کتابخانه های مختلف و مناسب با ماژول های مختلف وجود دارد که کدنویسی را راحت تر می کند. برای دریافت داده از ماهواره و ارسال آن به تلفن همراه، برنامه با استفاده از این نرم افزار نوشته می شود [۱۹].



شکل ۳-۹: نمایی از نرم افزار آردوینو

فصل چهارم

پیاده‌سازی سیستم ردیابی

در بخش قبلی معماری کلی سیستم و مراژول‌های موردنیاز برای پیاده‌سازی سیستم ردیابی را بیان و به صورت اجمالی معرفی کردیم. در این فصل به چگونگی قرار گرفتن و ارتباط بخش‌های مختلف می‌پردازیم و پیاده‌سازی سیستم را توضیح خواهیم داد.

سیستم ردیابی طراحی شده در این پروژه از مراژول آردوینو و سیم ۸۰۸ که شامل آنتن جی‌اس‌ام و جی‌پی‌اس می‌باشد، برای ردیابی استفاده می‌کند. هسته اصلی این پروژه میکروکنترلر آردوینو می‌باشد. موقعیت جغرافیایی شی تويیط آنتن جی‌پی‌اس دریافت شده و سپس این اطلاعات با استفاده از تکنولوژی جی‌اس‌ام به وب سرور فرستاده می‌شود. برای مشاهده کردن و ردیابی شی بر روی نقشه، یک برنامه کاربردی تحت وب توسعه داده شده است. این برنامه کاربردی از دو بخش فرانت‌اند و بکاند تشکیل شده است که قسمت فرانت‌اند^۱ با استفاده از فریمورک Angular و قسمت بکاند^۲ آن با استفاده از فریمورک Express توسعه داده شده است.

در ابتدا مراژول سیم ۸۰۸ برای گرفتن موقعیت مکانی از ماهواره مقداردهی اولیه می‌شود. تنظیمات اولیه این دستگاه با استفاده از دستورات AT انجام می‌شود. با متصل کردن آنتن چی‌پی‌اس این مراژول قادر خواهد بود مختصات مکانی را از ماهواره دریافت کند. سپس تنظیمات مربوط به شبکه جی‌پی‌آراس انجام می‌شود.

آنthen‌های جی‌اس‌ام و جی‌پی‌اس به مراژول سیم ۸۰۸ متصل می‌شوند. برد آردوینو و مراژول سیم ۸۰۸ دارای پین اتصال به زمین مشترک هستند. برنامه نوشته شده به زبان سی، با استفاده از نرم‌افزار Arduino IDE روی برد آردوینو آپلود می‌شود.

در شکل ۱-۴ نحوه اتصال مژول‌های مختلف در سیستم ردیابی نشان داده شده است:



شکل ۱-۴: سیستم ردیابی طراحی شده

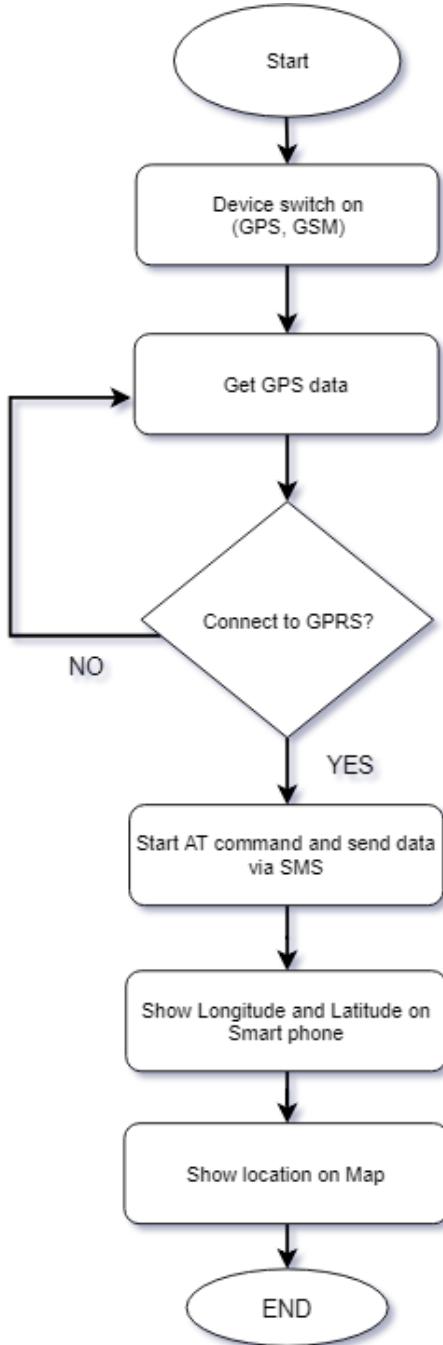
۱-۴ بررسی عملکرد سیستم ردیابی

همانطور که گفته شد، قسمت سخت افزاری سیستم ما از چهار مژول سیم، گیرنده جی‌اس‌ام، گیرنده جی‌پی‌اس و میکروکنترلر آردوینو تشکیل شده است. در این قسمت، پیاده‌سازی سیستم طراحی شده، شیوه ارتباط اجزای مختلف و کد پیاده‌سازی شده را توضیح خواهیم داد.

۱-۱-۴ بررسی عملکرد مدار

پیش از پرداختن به مژول‌ها و شیوه اتصال آن‌ها، لازم است شیوه عملکرد میکروکنترلر سیستم و نحوه پردازش اطلاعات را در فلوچارتی مشاهده کنیم. عملکرد کلی سیستم در بخش قبل توضیح داده شد. اکنون با ارئه فلوچارتی درباره الگوریتم پیاده‌سازی شده می‌توانیم درک بهتری نسبت به روند کار در مدار سخت افزاری طراحی شده و کد نوشته شده برای آن داشته باشیم.

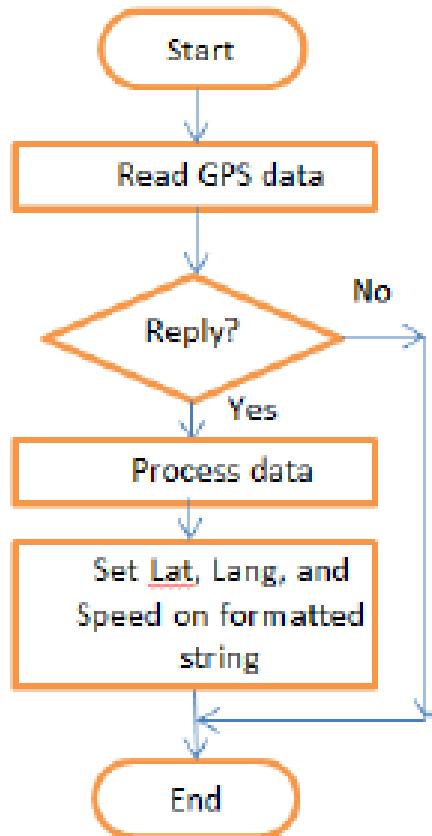
فلوچارت شکل ۲-۴ روند کلی کد پیاده‌سازی شده بر روی آردوینو را نمایش می‌دهد.



شکل ۲-۴: عملکرد کد پیاده‌سازی شده بر روی آردوینو [۱۹، ۱۱، ۱۰]

در ابتدای کار برای تست سیستم طراحی شده، آنتن جی‌پی‌اس به مازول سیم ۸۰۸ متصل می‌شود تا موقعیت مکانی (طول و عرض جغرافیایی) شی را از ماهواره دریافت کند. برای انجام این کار از نرم‌افزار Arduino IDE برای پروگرم کردن کد نوشته شده بر روی برد آردوینو استفاده می‌شود.

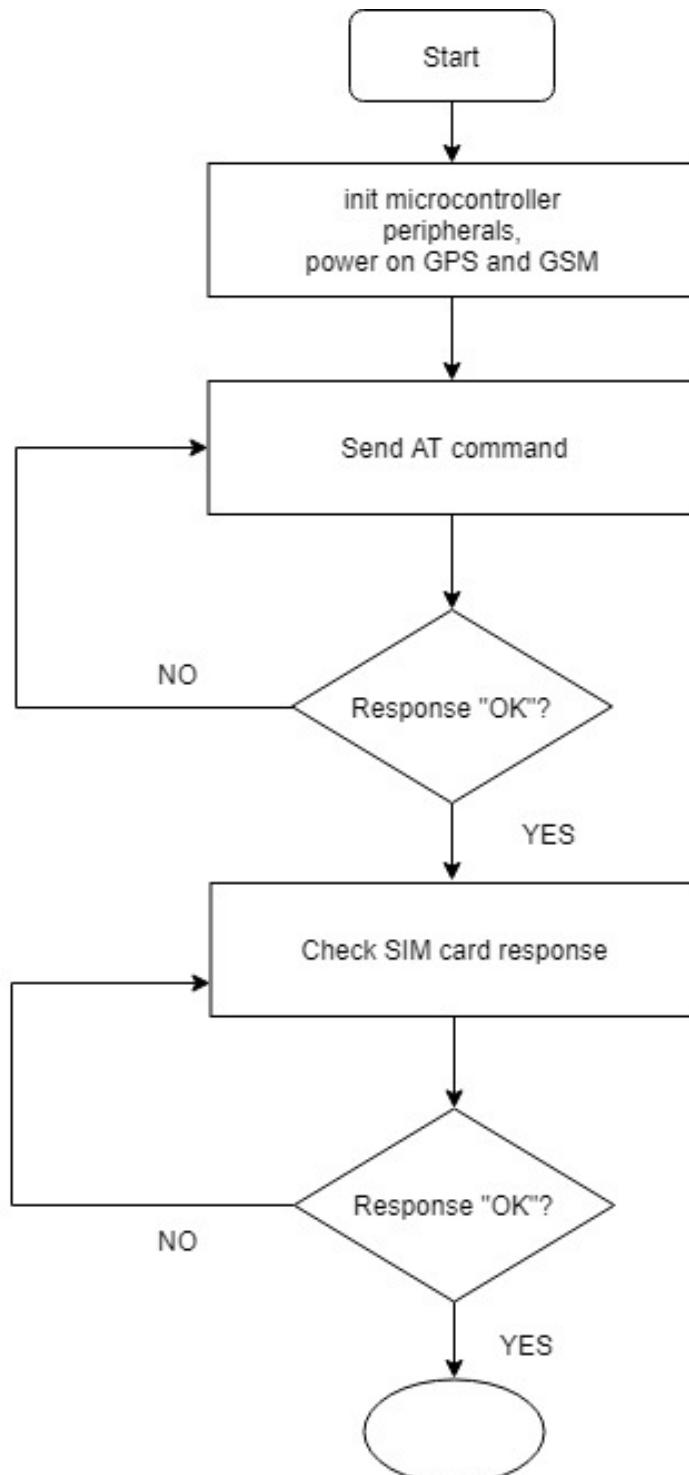
در فلوچارت شکل ۳-۴ نحوه کار جی‌پی‌اس نشان داده شده است.



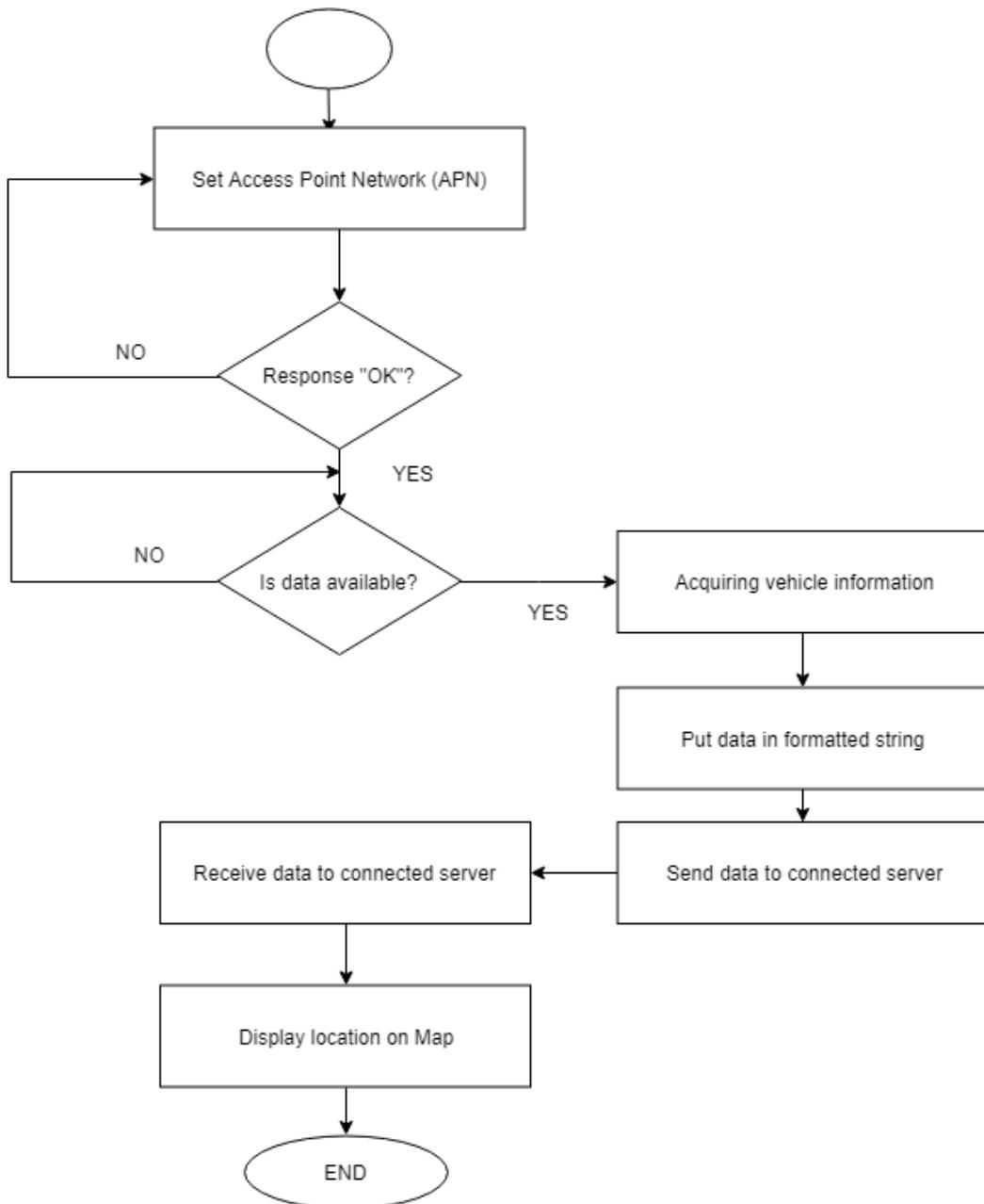
[A] شکل ۳-۴: فلوچارت خواندن اطلاعات GPS

برای ارسال موقعیت مکانی شی به کاربر از طریق شبکه جی‌اس‌ام از مازول سیم ۸۰۸ و میکروکنترلر آردوینو متصل به آن استفاده می‌شود. برای ارتباط مازول سیم ۸۰۸ با شبکه جی‌اس‌ام از دستورات AT برای برنامه‌نویسی و کنترل آن استفاده می‌کنیم.

در فلوچارت شکل ۴-۴ نحوه کار جی‌اس‌ام نشان داده شده است.



شکل ۴-۴: فلوچارت نحوه کار GSM [۸]



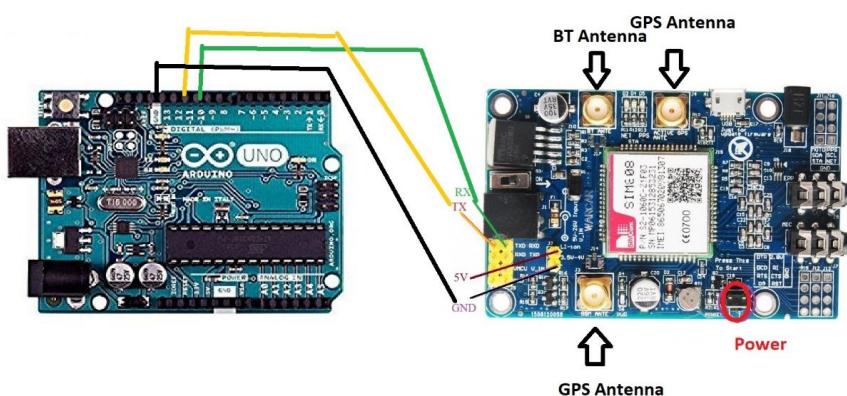
شکل ۴-۵: ادامه فلوچارت نحوه کار [A] GSM

سیستم ردیابی طراحی شده متشکل از ماژول سیم ۸۰۸ متصل به برد آردوینو می‌باشد. این ماژول دستور AT را می‌فرستد، اگر جواب این دستور OK باشد وضعیت شبکه چک می‌شود. بعد از چک شدن وضعیت شبکه و وصل شدن به آن، وضعیت جی‌پی‌اس چک شده و موقعیت مکانی شی از آن دریافت می‌شود. پس از دریافت اطلاعات جغرافیایی، درخواست ارسال اطلاعات به سرور ارسال می‌شود. در انتهای ارسال شده در سمت سرور پردازش شده و مسیر حرکت شی بر روی نقشه نشان داده خواهد شد.

۲-۴ بررسی معماری مدار

در این قسمت بخش سخت افزاری سیستم پیشنهادی را توضیح می‌دهیم. همانطور که در قسمت‌های قبل مشخص شد، با استفاده از آنتن جی پی اس متصل به مژول سیم ۸۰۸ اطلاعات مربوط به موقعیت مکانی، از هر دو دقیقه یکبار از ماهواره دریافت شده و سپس از طریق شبکه جی‌اس‌ام به سرور فرستاده می‌شود.

در شکل ۷-۴ شیوه اتصال مژول سیم ۸۰۸ به آردوبینو را مشاهده می‌کنید.



شکل ۷-۴: شیوه اتصال آردوبینو به مژول سیم ۸۰۸ [۳][۸۰۸]

ماژول سیم ۸۰۸ با استفاده از رابط سریال به آردوبینو متصل شده است، دارای دو پایه RX و TX است که به ترتیب به پایه دیجیتال ۱۰ و ۱۱ آردوبینو متصل شده‌اند. پایه اتصال به زمین این مژول هم به پایه اتصال به زمین برد آردوبینو متصل شده است. ولتاژ موردنیاز مژول سیم ۸۰۸ را از طریق آداپتور ۹ ولت خروجی تامین می‌کنیم.

۳-۴ برنامه کاربردی

در قسمت بکاند یک وب سرویس با معماری RESTful توسعه داده شده است که این وب سرویس داده‌های دریافتی را در پایگاه داده موردنظر ذخیره می‌کند و برنامه تحت وب این اطلاعات را دریافت کرده و نمایش می‌دهد.

REST یک معماری وب سرویس است که از HTTP برای تبادل اطلاعات میان دو سیستم استفاده می‌کند. ایده اصلی این معماری این است که به جای استفاده از مکانیزم‌های پیچیده برای اتصال ماشین‌ها

از HTTP برای برقراری اطلاعات بین ماشین‌ها استفاده کند. در این قسمت ابتدا ساختار پایگاه داده و شیوه ارتباط با آن را بیان می‌کنیم. در پایان نیز برنامه تحت وب نوشته شده را که مسیر حرکت شی را بر روی نقشه نمایش می‌دهد، را توضیح می‌دهیم.

۱-۳-۴ پایگاه داده

برای ذخیره اطلاعات و مدیریت سرور از پایگاه داده مانگو^۴ استفاده کرده‌ایم. مانگو دی‌بی یک پایگاه داده NOSQL است که روی سیستم عامل‌های مختلف از جمله ویندوز، مکینتاش و لینوکس اجرا می‌شود و همچنین اغلب زبان‌های برنامه‌نویسی را پشتیبانی می‌کند. مانگو دی‌بی کارایی بالا، دسترس پذیری، مقیاس پذیری، قابلیت تکرارهای سریع و اشتراک پذیری خودکار را فراهم می‌کند. مانگو دی‌بی به دلیل ساختار NOSQL تنها داده‌ها را ذخیره و جستجو می‌کند و در نتیجه سرعت دستیابی و ذخیره داده‌ها به شدت افزایش می‌یابد.

در پایگاه داده اطلاعات دریافتی از سمت سخت‌افزار، که هر دو دقیقه یکبار موقعیت شی از ماهواره دریافت می‌شود، را ذخیره می‌کنیم که این اطلاعات شامل طول و عرض جغرافیایی، سرعت، زمان و تاریخ است.

۲-۳-۴ نمایش اطلاعات

۱-۲-۳-۴ وب اپلیکیشن

برای نمایش اطلاعات ذخیره شده در پایگاه داده، یک وب اپلیکیشن توسعه داده شده است. قسمت فرانت‌اند این وب اپلیکیشن با استفاده از فریمورک Angular پیاده‌سازی شده و برای نمایش مسیر حرکت شی از Google Map استفاده می‌کنیم.

این وب اپلیکیشن از طریق آدرس زیر در دسترس است: <http://103.216.62.79> پردازش‌های سمت سرور با استفاده از فریمورک Express پیاده‌سازی شدند. نحوه عملکرد کد سمت سرور به این صورت است که به محض دریافت داده‌های جدید از سمت سخت‌افزار، این داده‌ها در پایگاه داده مانگو ذخیره می‌شوند. سخت‌افزار برای فرستادن داده جدید به سرور، هر بار یک درخواست HTTP به این شکل به سرور می‌فرستد:

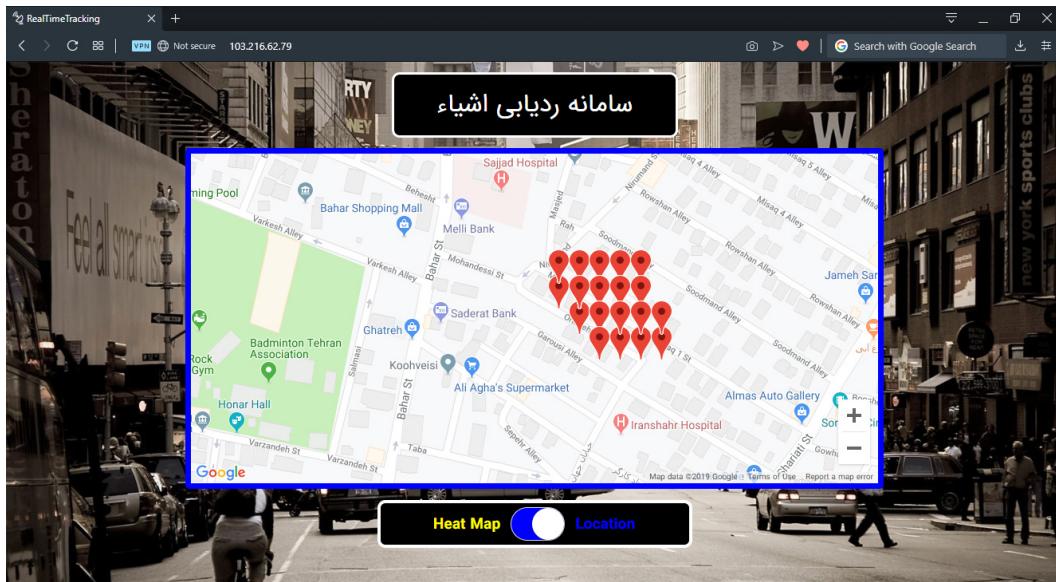
/api/device/add/

کد نوشته شده در سمت سرور به این صورت عمل می‌کند که هر ۶۰ ثانیه یکبار به پایگاه داده مراجعه می‌کند و با توجه به آخرین زمان دریافت اطلاعات از آن، اطلاعات دریافتی از آن زمان به بعد را دریافت

MongoDB^۵

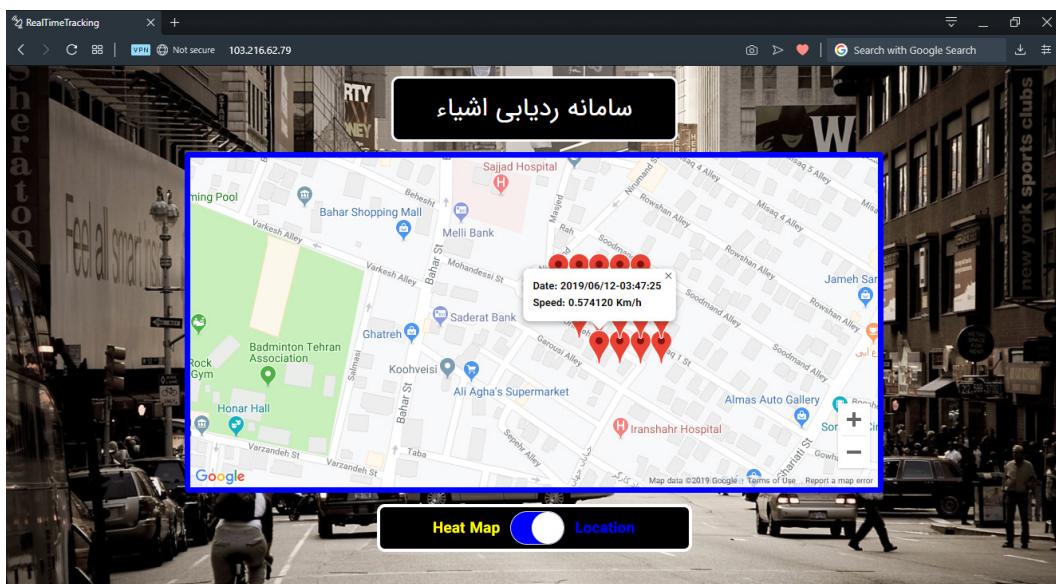
کرده و بدین ترتیب مسیر حرکت شی را بر روی نقشه نمایش می‌دهد.

شکل ۷-۴ برنامه کاربردی طراحی شده را نمایش می‌دهد.



شکل ۷-۴: نمایش مسیر حرکت شی در برنامه کاربردی

اگر بر روی هر کدام از مارکرهای موجود در نقشه که بیانگر موقعیت شی در هر لحظه می‌باشد، کلیک کنیم میتوان سرعت حرکت شی، زمان و تاریخ را در آن موقعیت را مشاهده خواهیم کرد. در شکل زیر نمایی از این خروجی را مشاهده می‌کنیم:



شکل ۷-۵: نمایش سرعت، زمان و تاریخ حرکت شی در هر موقعیت بر روی نقشه

۴-۳-۲-۲ پیام کوتاه

یکی از قابلیت‌های منحصر به فرد سیستم ردیابی پیاده‌سازی شده این است که میتوان به شماره سیم‌کارت درج شده بر روی مژول سیم ۸۰۸، برای دریافت موقعیت فعلی شی موردنظر، پیامی فرستاد. این مژول هم پس از دریافت پیام، موقعیت مکانی شی را از طریق جی‌پی‌اس دریافت کرده و اطلاعات لازم برای ردیابی را در پاسخ به پیام دریافتی ارسال می‌کند. این اطلاعات شامل طول و عرض جغرافیایی و سرعت شی می‌باشد. همچنین یک لینک هم به این پیام پیوست شده است که کاربر با کلیک بر روی آن می‌تواند موقعیت شی را بر روی نقشه مشاهده نماید.

در شکل‌های زیر پیام ارسال شده و پاسخ آن از طریق سیستم طراحی شده را مشاهده می‌کنید.

```

COM9
Open the GPS power success
Init Success, please send SMS message to me!
messageIndex: 18
From number: +989164450465
Datetime: 19/05/24,10:07:42+18
Recieved Message: Hi
2019/6/7 6:29:34:97
latitude :35.71
longitude :51.43
speed_kph :0.20
heading :200.64

Sim808 init success
Start to send message ...
Latitude : 35.71
Longitude : 51.43
Wind Speed : 0.20 kph
You can see your device in this location:
http://maps.google.com/maps?q= 35.71, 51.43

+989164450465

```

The terminal window has several controls at the bottom: 'Autoscroll' (checked), 'Show timestamp' (unchecked), 'Both NL & CR' (selected), '9600 baud' (selected), and 'Clear output'.

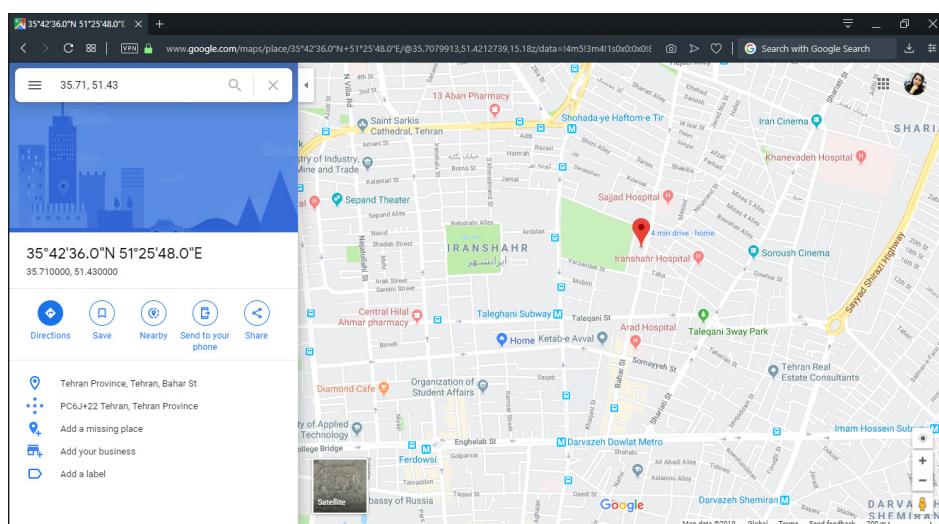
شکل ۹-۴: دریافت پیام ارسالی کاربر برای دریافت موقعیت توسط مژول سیم ۸۰۸

شکل ۴-۴ پیام ارسال شده برای کاربر را نشان می‌دهد.



شکل ۴-۴: پیام دریافت شده توسط کاربر

با کلیک بر لینک دریافت شده میتوان موقعیت فعلی شی را بر روی نقشه مشاهده نمود.



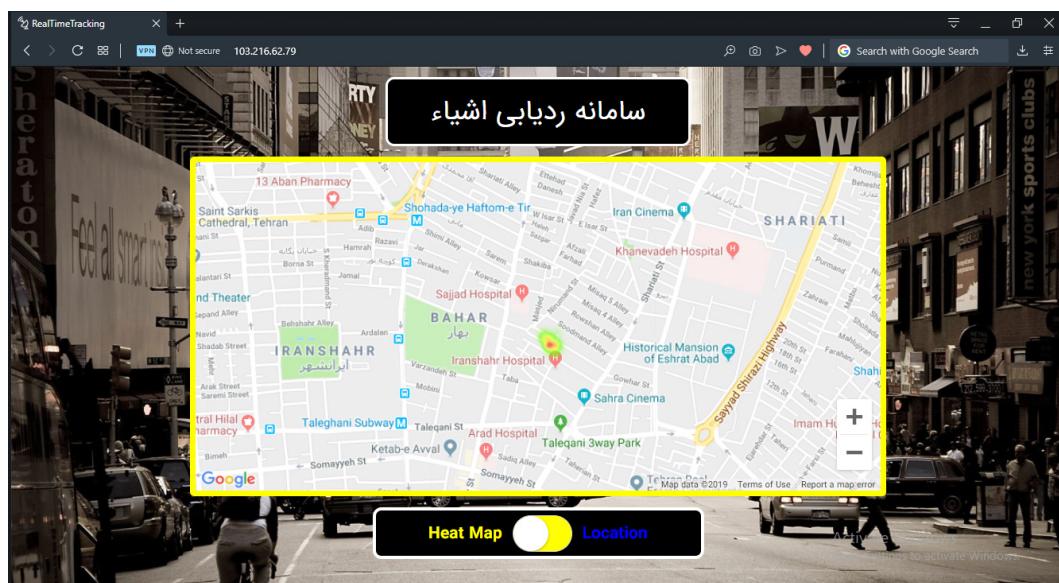
شکل ۴-۱۱: مشاهده موقعیت شی بر روی نقشه

۴-۴ تحلیل اطلاعات

با استفاده از سیستم ردیابی بی‌درنگ طراحی شده، توانستیم موقعیت مکانی شی رو به طور پیوسته اندازه‌گیری کرده و این اطلاعات را در پایگاه داده ذخیره کنیم. یکی دیگر از قابلیت‌های این سیستم بدست آوردن مکان‌های پر تردد فرد در یک بازه زمانی مشخص است. برای این کار از هیت مپ (نقشه حرارتی)^۴ استفاده کردیم.

هیت مپ داده‌های گرافیکی و رنگی هستند که ما را قادر می‌سازند تا رفتار کاربرانمان را شناسایی کنیم. در این پروژه، هیت مپ را با استفاده از اطلاعات ذخیره شده که در واقع موقعیت‌های مکانی شی دز زمان‌های مختلف بوده است، پیاده سازی کردیم. مکان‌های پر تردد فرد که فراوانی آن‌ها در مجموعه داده ما بیشتر از بقیه بوده، به صورت نواحی قرمز رنگ بر روی نقشه قابل مشاهده خواهند بود.

برنامه کاربردی طراحی شده شامل دو نقشه است. به این صورت است که در ابتدا موقعیت‌های شی را بر روی نقشه نشان می‌دهد و با انتخاب گزینه هیت‌مپ می‌توان هیت مپ این نقاط را مشاهده نمود. در شکل ۱۲-۴ بر روی هیت مپ مکان‌های پر تردد شی که به صورت نواحی قرمزرنگ نشان داده است را مشاهده می‌کنید.



شکل ۱۲-۴: مکان‌های پر تردد شی بر روی هیت مپ

فصل پنجم

جمع‌بندی و کارهای آتی

۱-۵ جمع‌بندی

در این پروژه هدف ما پیاده‌سازی یک سیستم ردیابی بی‌درنگ برای ردیابی اشیا متحرک بوده است. لذا در این پروژه سیستم ردیابی، برای مانیتور کردن موقعیت شی متحرک از طریق پیام کوتاه و همچنین به صورت آنلاین بر روی نقشه تست و پیاده‌سازی نمودیم. هسته اصلی سیستم طراحی شده، برد آردوینو و ماژول سیم ۸۰۸ می‌باشد. مودم جی‌اس‌ام با استفاده از دستورات AT کنترل شده و از این طریق امکان تبادل اطلاعات با استفاده از شبکه جی‌اس‌ام را فراهم می‌آورد. در این پروژه برای یافتن موقعیت شی از ماژول جی‌پی‌اس استفاده شده است. جی‌پی‌اس هر دو دقیقه یکبار موقعیت مکانی شی را از ماهواره دریافت کرده و این اطلاعات به سمت سرور فرستاده می‌شوند. سرور اطلاعات دریافتی را برای وب سرویس پیاده‌سازی شده، ارسال می‌کند. در نهایت این داده‌ها در پایگاه داده ذخیره شده و در برنامه کاربردی نمایش داده می‌شوند و میتوانیم مسیر حرکت شی را بر روی نقشه مشاهده کنیم.

برنامه پیاده‌سازی شده بر روی آردوینو به این صورت نوشته شده است که هر دو دقیقه یکبار موقعیت مکانی شی را با استفاده از جی‌پی‌اس دریافت می‌کند و به سرور می‌فرستد. در سمت سرور این اطلاعات در پایگاه داده مانگو ذخیره شده و سپس با استفاده از این اطلاعات مسیر حرکت شی بر روی نقشه داده می‌شود. کد سمت سرور به این صورت نوشته شده است که هر یک دقیقه یکبار به پایگاه داده مراجعه کرده و نقشه حاوی مسیر حرکت شی به‌روزرسانی می‌شود.

در پایان این پروژه توانستیم سیستم ردیابی بی‌درنگی را طراحی کنیم که امروزه دارای کاربرد فراوان در زمینه‌های مختلف مانند ردیابی وسایل نقلیه، کودکان و سالمندان و ... می‌باشد و با استفاده از آن قادر خواهیم بود اقدامات لازم را در سریع‌ترین زمان ممکن انجام دهیم.

۲-۵ کارهای آتی

ماژول جی‌پی‌اس توان مصرفی نسبتاً بالایی دارد. دو راه پیشنهادی برای کاهش توان مصرفی در این سیستم وجود دارد. اولین راهی که میتوان برای این سیستم پیشنهاد داد استفاده از الگوریتم‌های یادگیری ماشین برای پیش‌بینی حرکت جسم است. روش دیگر عدم استفاده از جی‌پی‌اس و استفاده از پروتکل LoRaWAN برای یافتن موقعیت سیستم است.

یکی دیگر از کارهایی که در ادامه این پروژه می‌توان انجام داد، اضافه کردن حسگرها و عملگرها به این سیستم است. به عنوان مثال از سنسور شتاب‌سنج میتوان استفاده نمود که حرکت سیستم را

بررسی کنیم و تنها زمانی که سیستم در حال حرکت باشد، اطلاعات جی‌پی‌اس به سرور فرستاده شود که این کار موجب کاهش توان مصرفی خواهد شد.

همچنین میتوان بر روی سیستم طراحی شده یک دوربین هم نصب نمود تا علاوه بر اطلاعات ارسالی از طریق جی‌پی‌اس، بتوانیم با استفاده از تصاویر این دوربین هم موقعیت مکانی را رصد کنیم.

همانطور که گفتیم یکی از قابلیت‌های این سیستم دریافت موقعیت فرد با ارسال پیام به شماره سیم‌کارت درج شده در این سیستم بود که موجب می‌شد این قابلیت علی‌رغم مزیت‌هایی که دارد، خطرات امنیتی را برای افراد ایجاد کند. لذا میتوان امنیت سیستم ردبایی موردنظر را بهبود بخشد و مشخص نمود در صورت دریافت پیام از شماره مشخص و از پیش تعیین شده، موقعیت فرد را ارسال کند و در غیر این صورت هیچ‌گونه اطلاعاتی مبادله نشود.

این سیستم به‌گونه‌ای طراحی شده است که میتوان آن را برای کاربردهای مختلف شخصی‌سازی نمود. به عنوان مثال با مشخص کردن محدوده مجاز برای حرکت شی، میتوان سیستم را به‌گونه‌ای طراحی کرد که به محض خروج از محدوده مشخص شده، سیستم به صورت خودکار با نهادهایی مثل پلیس و ... تماس برقرار کند و خطاهای انسانی را به حداقل برساند.

مراجع و مراجع

- [1] <https://www.commsbusiness.co.uk/features/what-are-the-internet-of-things-and-m2m/>.
- [2] <https://www.slideshare.net/sumitcan/iot-architecture>.
- [3] <https://maker.pro/arduino/projects/build-a-car-tracking-system-with-the-sim808-module>.
- [4] [https://cdn-shop.adafruit.com/datasheets/SIM808_Hardware + DesignV1.00.pdf](https://cdn-shop.adafruit.com/datasheets/SIM808_Hardware+DesignV1.00.pdf).
- [5] Agrawal, Tarun and Qadeer, Mohamaad Abdul. Tracing path with arduino uno using GPS and GPRS/GSM. in *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*. IEEE, September 2018.
- [6] Bidabad, Behroz and Tayebi, Akbar. Design and implementation of vehicle tracking system using gps/gsm/gprs technology and smartphone application. *arXiv preprint arXiv:0710.2816*, 2014.
- [7] Cui, Youjing and Ge, Shuzhi Sam. Autonomous vehicle positioning with gps in urban canyon environments. *IEEE Transactions on Robotics and Automation*, February 2003.
- [8] ElShafee, Ahmed, Menshawi, Mahmoud El, and Saeed, Mena. Integrating social network services with vehicle tracking technologies. *International Journal of Computer Applications*, June 2013.
- [9] Falih, Muntasser and Salim, Khalifaa. *Design and Implementation of GPS Based Navigation System*. Ph.D. thesis, September 2013.

- [10] Hazza Alshamisi, Veton Këpuska. Real time gps vehicle tracking system. *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, March 2017.
- [11] Humaid Alshamsi, Veton Këpuska, Hazza Alshamsi. Real time vehicle tracking using arduino mega. *International Journal of Science and Technology*, December 2016.
- [12] Koyuncu, Baki and Özdemir, Zeynep. Real time position detection by using gps+gsm+gprs and arduino mega based telit gl865. *International Journal of Electronics Communication and Computer Engineering*, December 2016.
- [13] Madakam, Somayya, Ramaswamy, R., and Tripathi, Siddharth. Internet of things (IoT): A literature review. *Journal of Computer and Communications*, 2015.
- [14] Mahamulkar, Snehal Uttam and Yawale, Prof. R. U. Design and development of vehicle tracking and monitoring system. in *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*. IEEE, September 2017.
- [15] Mangla, Neha, Sivananda, G, Kashyap, Aishwarya, and Vinutha. A GPS-GSM predicated vehicle tracking system, monitored in a mobile app based on google maps. in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. IEEE, August 2017.
- [16] Mohamad, Omar Abdulwahabe, Hameed, Rasha Talal, and Tapus, Nicolae. Design and implementation of real time tracking system based on arduino intel galileo. in *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. IEEE, June 2016.
- [17] Mukhtar, Mashhood. GPS based advanced vehicle tracking and vehicle control system. *International Journal of Intelligent Systems and Applications*, February 2015.

- [18] Pham, Hoang Dat, Drieberg, Micheal, and Nguyen, Chi Cuong. Development of vehicle tracking system using GPS and GSM modem. in *2013 IEEE Conference on Open Systems (ICOS)*. IEEE, December 2013.
- [19] Rahman, Md. Marufi, Mou, Jannatul Robaiat, Tara, Kusum, and Sarkar, Md. Ismail. Real time google map and arduino based vehicle tracking system. in *2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*. IEEE, December 2016.
- [20] Saranya, J. and Selvakumar, J. Implementation of children tracking system on android mobile terminals. in *2013 International Conference on Communication and Signal Processing*. IEEE, April 2013.
- [21] Shah, Sajjad Hussain and Yaqoob, Ilyas. A survey: Internet of things (IOT) technologies, applications and challenges. in *2016 IEEE Smart Energy Grid Engineering (SEGE)*. IEEE, August 2016.
- [22] Song, H., Zhu, S., and Cao, G. SVATS: A sensor-network-based vehicle anti-theft system. in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*. IEEE, April 2008.

پیوست

۱. کد پیاده‌سازی شده بر روی آردوینو

```
1 //*****
2 * Bsc Final Project
3 * Real Time Tracking system
4 * Sareh Soltani Nejad, Email: sare.soltani74@gmail.com / sare1996@aut.ac.ir
5 * 13/june/2019
6 *****/
7
8 #include<SoftwareSerial.h>
9 extern uint8_t SmallFont[];
10
11 #define rxPin 11
12 #define txPin 10
13
14 SoftwareSerial mySerial(txPin, rxPin);
15
16 const char url[] = "http://103.216.62.79/api/device/add/";
17 char response[200];
18 char latitude[15];
19 char longitude[15];
20 char altitude[16];
21 char date[24];
22 char TTFF[3];
23 char satellites[3];
24 char speedOTG[10];
25 char course[15];
26
27 void setup(){
28     mySerial.begin(9600);
29     Serial.begin(9600);
30     Serial.println("Starting...");
31     power_on();
```

```

32     start_GPS();
33     while (sendATcommand("AT+CREG?", "+CREG: 0,1", 2000) == 0);
34     sendATcommand("AT+SAPBR=3,1,\"Contype\"", "GPRS\"", "OK", 2000);
35     sendATcommand("AT+SAPBR=3,1,\"APN\"", "mtnirancell\"", "OK", 2000);
36     sendATcommand("AT+SAPBR=3,1,\"USER\"", "\",\"", "OK", 2000);
37     sendATcommand("AT+SAPBR=3,1,\"PWD\"", "\",\"", "OK", 2000);
38     // gets the GPRS bearer
39     sendATcommand("AT+SAPBR=0,1", "OK", 20000);
40     while (sendATcommand("AT+SAPBR=1,1", "OK", 20000) == 0){
41         delay(5000);
42     }
43 }
44
45 void loop(){
46     // gets GPS data
47     get_GPS();
48     // sends GPS data to the script
49     send_HTTP();
50     //sendNMEALocation("989164450465",frame);
51     delay(12000);
52 }
53
54 void power_on(){
55     uint8_t answer=0;
56     // checks if the module is started
57     answer = sendATcommand("AT", "OK", 2000);
58     if (answer == 0){
59         // waits for an answer from the module
60         while(answer == 0){
61             // Send AT every two seconds and wait for the answer
62             answer = sendATcommand("AT", "OK", 2000);
63         }
64     }

```

```

65     Serial.println("Power on");
66 }
67
68 int8_t start_GPS(){
69     // starts the GPS
70     while(sendATcommand("AT+CGNSPWR=1", "OK", 2000)==0);
71     while(sendATcommand("AT+CGPSRST=0", "OK", 2000)==0);
72     // waits for fix GPS
73     while((
74         sendATcommand("AT+CGPSSTATUS?", "2D Fix", 5000) ||
75         sendATcommand("AT+CGPSSTATUS?", "3D Fix", 5000)) == 0 ) );
76     return 1;
77 }
78
79 int8_t get_GPS(){
80     int8_t answer;
81     char * auxChar;
82     // request Basic string
83     sendATcommand("AT+CGPSINF=0", "O", 8000);
84     auxChar = strstr(response, "+CGPSINF:");
85     if (auxChar != NULL){
86         // Parses the string
87         memset(longitude, '\0', 15);
88         memset(latitude, '\0', 15);
89         memset(alitude, '\0', 16);
90         memset(date, '\0', 24);
91         memset(TTFF, '\0', 3);
92         memset(satellites, '\0', 3);
93         memset(speedOTG, '\0', 10);
94         memset(course, '\0', 15);
95         strcpy (response, auxChar);
96         Serial.println(response);
97     }

```

```

98     strtok(response, ",");
99     strcpy(latitude,strtok(NULL, ",")); // Gets longitude
100    strcpy(longitude,strtok(NULL, ",")); // Gets latitude
101    strcpy(altitude,strtok(NULL, ",")); // Gets altitude
102    strcpy(date,strtok(NULL, ",")); // Gets date
103    strcpy(TTFF,strtok(NULL, ","));
104    strcpy(satellites,strtok(NULL, ",")); // Gets satellites
105    strcpy(speedOTG,strtok(NULL, ",")); // Gets speed over ground. Unit is knots.
106    strcpy(course,strtok(NULL, "\r")); // Gets course
107    convert2Degrees(latitude);
108    convert2Degrees(longitude);
109    answer = 1;
110 }
111 else
112     answer = 0;
113 return answer;
114 }
115
116 void sendNMEAlocation(char * cellPhoneNumber, char * message) {
117     char ctrlZString[2];
118     char sendSMSString[100];
119     // Started sendNMEAlocation.
120     memset(ctrlZString, '\0', 2);
121     ctrlZString[0] = 26;
122     memset(sendSMSString, '\0', 100);
123     sprintf(sendSMSString,"AT+CMGS=\"%s\"",cellPhoneNumber);
124     // request Basic string
125     sendATcommand(sendsSMSString, ">", 2000);
126     mySerial.println(message);
127     sendATcommand(ctrlZString, "OK", 6000);
128     //Ended sendNMEAlocation.
129 }
```

```

131 int8_t send_HTTP(){
132     int8_t answer;
133     char aux_str[200];
134     char frame[200];
135     // Initializes HTTP service
136     answer = sendATcommand("AT+HTTPINIT", "OK", 10000);
137     if (answer == 1){
138         // Sets CID parameter
139         answer = sendATcommand("AT+HTTPPARA=\"CID\",1", "OK", 5000);
140         if (answer == 1){
141             // Sets url
142             memset(aux_str, '\0', 200);
143             sprintf(aux_str, "AT+HTTPPARA=\"URL\", \"%s\", url");
144             mySerial.print(aux_str);
145             Serial.println(aux_str);
146             sprintf(frame, "?visor=false&lat=%s&lon=%s&alt=%s&date=%s&TTFF=%s&sat=%s&speedOTG=%s&course=%s"
147             , latitude, longitude, altitude, date, TTFF, satellites, speedOTG, course);
148             Serial.println(frame);
149             mySerial.print(frame);
150             answer = sendATcommand("", "OK", 5000);
151             if (answer == 1){
152                 // Starts GET action
153                 answer = sendATcommand("AT+HTTPACTION=0", "+HTTPACTION: 0,200", 30000);
154                 if (answer == 1){
155                     Serial.println(F("Done!"));
156                 }
157                 else{
158                     Serial.println(F("Error getting url"));
159                 }
160             }
161             else{
162                 Serial.println(F("Error setting the url"));
163             }
164 }
```

```

164     }
165     else{
166         Serial.println(F("Error setting the CID"));
167     }
168     else{
169         Serial.println(F("Error initializing"));
170     }
171
172     sendATcommand("AT+HTTPTERM", "OK", 5000);
173     return answer;
174 }
175
176 int8_t sendATcommand(char* ATcommand, char* expected_answer1, unsigned int timeout){
177     uint8_t x=0, answer=0;
178     unsigned long previous;
179     char readVar[200];
180     char * auxChar;
181
182     memset(response, '\0', 200); // Initialize the string
183     memset(readVar, '\0', 200); // Initialize the string
184
185     while( mySerial.available() > 0) mySerial.read(); // Clean the input buffer
186     while( Serial.available() > 0) Serial.read(); // Clean the input buffer
187
188     mySerial.write(ATcommand); // Send the AT command
189     mySerial.write("\r\n\r\n"); // Send enter
190
191     Serial.println(ATcommand);
192
193     x = 0;
194     previous = millis();
195

```

```

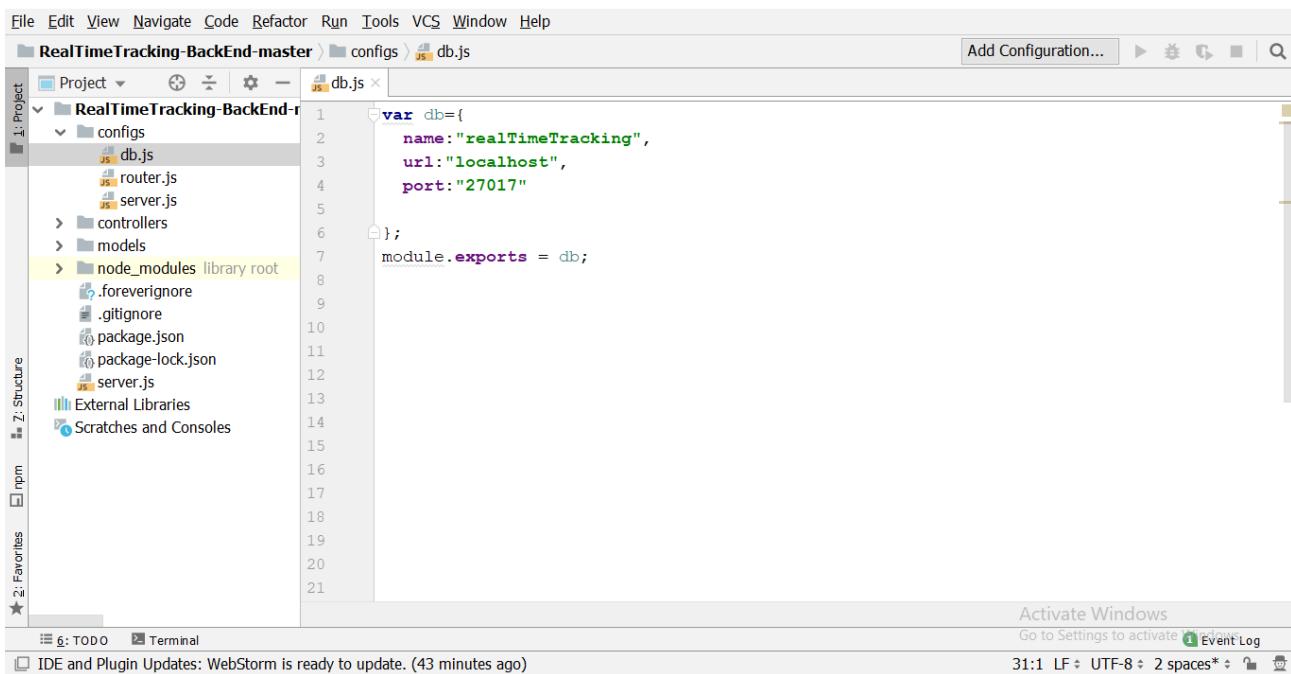
196     // this loop waits for the answer
197     do{
198         if(mySerial.available() != 0){
199             readVar[x] = mySerial.read();
200             x++;
201             // check if the desired answer is in the response of the module
202             auxChar = strstr(readVar, expected_answer1);
203             if (auxChar != NULL){
204                 if( strstr(readVar, "+CGPSINF:") == NULL)
205                     strcpy (response, auxChar);
206                 else
207                     strcpy (response, readVar);
208                 answer = 1;
209             }
210         }
211         // Waits for the asnwer with time out
212     }
213     while((answer == 0) && ((millis() - previous) < timeout));
214     if(auxChar == NULL)
215         Serial.println(readVar);
216     return answer;
217 }
218
219 void convert2Degrees(char* input){
220     char res[40];
221     float deg;
222     float minutes;
223     float seconds;
224     char aux[10] ="";
225     //latitude format: DDmm.mmummm'
226
227     // get 'degrees' from input parameter

```

```

227 // get 'degrees' from input parameter
228 aux[0] = input[0];
229 aux[1] = input[1];
230 aux[2] = '\0';
231
232 // convert string to integer and add it to final float variable
233 deg = atoi(aux);
234 // get 'minutes' from input parameter
235 for ( int i=0; i<7; i++ ){
236     aux[i] = input[i+2];
237 }
238 aux[7] = '\0';
239 // convert string to integer and add it to final float variable
240 minutes = atoi(aux);
241
242 // get 'seconds' from input parameter
243 for ( int i=0; i<2; i++ ){
244     aux[i] = aux[i+3];
245 }
246 aux[3] = '\0';
247 seconds = atoi(aux);
248
249 // add minutes to degrees
250 deg = deg + minutes/60 + seconds/3600;
251 dtostrf(deg, 2, 10, res);
252 strncpy(input, res, 9);
253 }
```

۲. کد پیاده‌سازی شده سمت بک‌اند



```
var router = {
  controllers : "../controllers",
  apiPrefix : "/api",
  devicePrefix: "/device",
  userPrefix: "/user",
  add:function(app){
    var path = require('path');
    app.use(this.apiPrefix+this.devicePrefix,require(path.join(this.controllers,'deviceController')));
    app.use(this.apiPrefix+this.userPrefix,require(path.join(this.controllers,'userController')));
  }
};
module.exports = router;
```

The screenshot shows the WebStorm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The title bar indicates the project is "RealTimeTracking-BackEnd-master" and the current file is "configs/server.js". A toolbar on the right contains icons for Add Configuration..., Run, Stop, Refresh, and Search. The left sidebar has sections for Project, Favorites, Structure, npm, and Scratches and Consoles. The main workspace shows the file "server.js" with the following code:

```
var path = require('path');
var server = {
  portNumber: 8080,
  publicDirectoryLocation: path.join(__dirname, '../public'),
};
module.exports = server;
```

The screenshot shows the WebStorm IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help.
- Title Bar:** RealTimeTracking-BackEnd-master > controllers > deviceController.js
- Add Configuration... button:** Located in the top right corner.
- Project Tree (Left):**
 - Project
 - RealTimeTracking-BackEnd-1
 - configs
 - controllers
 - deviceController.js
 - userController.js
 - models
 - node_moduleslibrary root
 - .foreverignore
 - .gitignore
 - package.json
 - package-lock.json
 - server.js
 - External Libraries
 - Scratches and Consoles
- Code Editor (Right):** The file deviceController.js is open, showing code related to location tracking using Express, fs, path, moment, and moment-timezone modules. It handles POST requests to '/add' to create a new location entry in a database model.
- Sidebar (Bottom Left):**
 - Z: Structure
 - npm
 - Favorites
- Bottom Status Bar:** Activate Windows, Go to Settings to activate, Event Log.
- Bottom Icons:** TODO, Terminal, IDE and Plugin Updates: WebStorm is ready to update. (49 minutes ago), 26:1 LF+, UTF-8+, 2 spaces*, Event Log.

The screenshot shows the WebStorm IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help.
- Title Bar:** RealTimeTracking-BackEnd-master > controllers > UserController.js
- Add Configuration...:** A button in the top right corner.
- Project Tree:** Shows the project structure: RealTimeTracking-BackEnd->controllers->UserController.js.
- Code Editor:** The file UserController.js contains the following code:

```
const express = require('express');
const fs = require('fs');
const path = require('path');
var locationModel = require('../models/Location');
var router = express.Router();

router.post("/mapCenter", (req, res) => {
  var center = {
    lat: 35.706421,
    lon: 51.426771
  };
  return res.status(200).json(center);
});

router.post("/mapData", (req, res) => {
  var lastDate = new Date(value: 'January 1, 1970 00:00:00');
  if(req.body.lastDate)
  {
    var userlastDate = Date.parse(req.body.lastDate);
    if(!isNaN(userlastDate))
      lastDate = userlastDate;
  }
  locationModel.find({date: {$gt: lastDate}}).sort({date:1}).exec(function(error, locations){
    if(error)
      return res.sendStatus(statusCode: 500);
    return res.status(200).json(locations);
  });
});
```

The code handles POST requests to '/mapCenter' and '/mapData'. The '/mapCenter' endpoint returns a JSON object representing a map center with latitude 35.706421 and longitude 51.426771. The '/mapData' endpoint finds documents in the 'locationModel' collection where the 'date' field is greater than a specified date ('January 1, 1970 00:00:00') and sorts them by 'date'. It then returns the results as a JSON array.

The screenshot shows the WebStorm IDE interface with the following details:

- File Path:** RealTimeTracking-BackEnd-master/controllers/userController.js
- Code Editor Content:**

```

11         return res.status(200).json(locations);
12     });
13     router.post("/mapData", (req, res) => {
14         var lastDate = new Date(value: 'January 1, 1970 00:00:00');
15         if (req.body.lastDate)
16         {
17             var userlastDate = Date.parse(req.body.lastDate);
18             if (!isNaN(userlastDate))
19                 lastDate = userlastDate;
20         }
21         locationModel.find({date: {$gt: lastDate}}).sort({date: 1}).exec(function(error, locations) {
22             if (error)
23                 return res.sendStatus(statusCode: 500);
24             return res.status(200).json(locations);
25         });
26     });
27     module.exports = router;
28 };
29 
```
- Project Explorer:** Shows the project structure with files like deviceController.js, Location.js, and server.js.
- Status Bar:** Shows "Activate Windows", "Event Log", "1:36 LF", "UTF-8", "2 spaces", and icons for TODO, Terminal, and IDE updates.

The screenshot shows the WebStorm IDE interface with the following details:

- File Path:** RealTimeTracking-BackEnd-master/models/Location.js
- Code Editor Content:**

```

1 var mongoose = require('mongoose');
2 var locationSchema = new mongoose.Schema({
3     lat: String,
4     lon: String,
5     alt: String,
6     speed: String,
7     date: Date
8 });
9 mongoose.model(name: 'Location', locationSchema);
10
11 module.exports = mongoose.model(name: 'Location');
12 
```
- Project Explorer:** Shows the project structure with files like deviceController.js, Location.js, and server.js.
- Status Bar:** Shows "Activate Windows", "Event Log", "12:1 LF", "UTF-8", "4 spaces", and icons for TODO, Terminal, and IDE updates.

۳. کد پیاده‌سازی شده سمت فراتر اند

```

<div class="title">
  <div class="banner">
    {{ title }}
  </div>
</div>
<div class="mapContainer">
  <agm-map class="map" [streetViewControl]="false" [latitude]="mapCenterLat"
    [longitude]="mapCenterLng" [zoom]="15"
    [ngClass]="'locationBorder':showLocation , 'heatMapBorder':!showLocation"
    (mapReady)="onMapLoad($event)">
    <div *ngIf="showLocation">
      <agm-marker *ngFor="let marker of mapData" [latitude]="marker.lat"
        [longitude]="marker.lon">
        <agm-info-window>
          <strong>Date: {{marker.date}}</strong><br/>
          <strong>Speed: {{marker.speed}} Km/h</strong>
        </agm-info-window>
      </agm-marker>
    </div>
  </agm-map>
</div>
<div class="buttonContainer">
  <span class="switchLabel" style="...>Heat Map</span>
  <ui-switch class="switch" [(ngModel)]="showLocation" (change)="toggleLocation()">
    <span class="switchLabel" style="...>Location</span>
  </ui-switch>
</div>

```

Activate Windows
Go to Settings to activate Event Log
3:16 LF + UTF-8 + 2 spaces* + 🔍

TSLint is enabled: 'tslint' is listed in package.json. // Disable TSLint (25 minutes ago)

```

<div *ngIf="showLocation">
  <agm-marker *ngFor="let marker of mapData" [latitude]="marker.lat"
    [longitude]="marker.lon">
    <agm-info-window>
      <strong>Date: {{marker.date}}</strong><br/>
      <strong>Speed: {{marker.speed}} Km/h</strong>
    </agm-info-window>
  </agm-marker>
</div>
</agm-map>
</div>
<div class="buttonContainer">
  <span class="switchLabel" style="...>Heat Map</span>
  <ui-switch class="switch" [(ngModel)]="showLocation" (change)="toggleLocation()">
    <span class="switchLabel" style="...>Location</span>
  </ui-switch>
</div>

```

Activate Windows
Go to Settings to activate Event Log
3:16 LF + UTF-8 + 2 spaces* + 🔍

TSLint is enabled: 'tslint' is listed in package.json. // Disable TSLint (26 minutes ago)

```

1 import { Injectable } from '@angular/core';
2 import { HttpClient, HttpHeaders } from '@angular/common/http';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class Data ApiService {
8   prefix:string = "/api/user"
9   constructor(private http: HttpClient) {
10 }
11   getMapCenter() {
12     return this.http.post<any>( url: this.prefix+"/mapCenter", body: null);
13   }
14   getMapData(date) {
15     return this.http.post<any>( url: this.prefix+"/mapData", body: {lastDate:date});
16   }
17 }
18
19
20
21

```

Activate Windows
Go to Settings to activate Event Log

1:1 LF + UTF-8 + 2 spaces*

TSLint is enabled: 'tslint' is listed in package.json. // Disable TSLint (29 minutes ago)

```

1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <title>RealTimeTracking</title>
6     <base href="/">
7
8     <meta name="viewport" content="width=device-width, initial-scale=1">
9     <link rel="icon" type="image/x-icon" href="favicon.ico">
10
11   </head>
12   <body>
13     <app-root></app-root>
14   </body>
15

```

Activate Windows
Go to Settings to activate Event Log

15:1 LF + UTF-8 + 2 spaces*

TSLint is enabled: 'tslint' is listed in package.json. // Disable TSLint (35 minutes ago)

۴. کد پیاده‌سازی بر روی آردوینو برای ارسال و دریافت پیام کوتاه

```

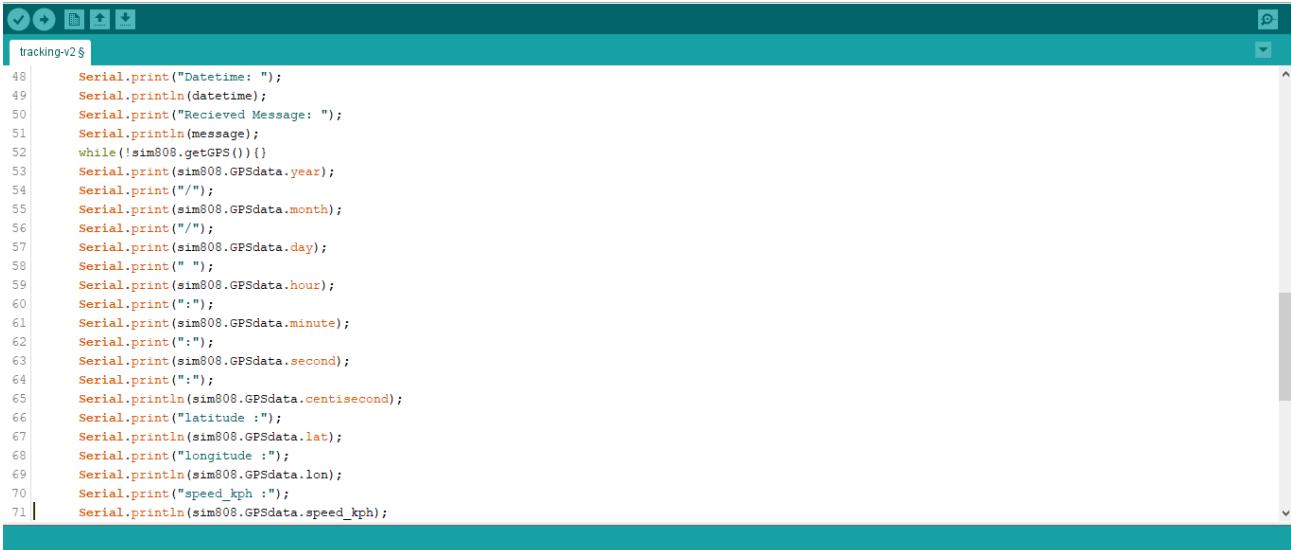
tracking-v2.ino
1 #include <sim808.h>
2 #include <DFRobot_sim808.h>
3 #include <SoftwareSerial.h>
4 #define MESSAGE_LENGTH 160
5 char message[MESSAGE_LENGTH];
6 int messageIndex = 0;
7 char MESSAGE[300];
8 char lat[12];
9 char lon[12];
10 char wspeed[12];
11 char phone[16];
12 char datetime[24];
13
14 #define PIN_TX 10
15 #define PIN_RX 11
16
17 SoftwareSerial mySerial(PIN_TX,PIN_RX);
18 DFRobot_SIM808 sim808(&mySerial); //Connect RX,TX,PWR,
19
20 void setup(){
21   mySerial.begin(9600);
22   Serial.begin(9600);
23   //***** Initialize sim808 module *****
24   while(!sim808.init()){


```

```

tracking-v2.ino
25   Serial.print("Sim808 init error\r\n");
26   delay(1000);
27 }
28 delay(3000);
29 if( sim808.attachGPS() )
30   Serial.println("Open the GPS power success");
31 else
32   Serial.println("Open the GPS power failure");
33 Serial.println("Init Success, please send SMS message to me!");
34 }
35
36 void loop(){
37   //***** Detecting unread SMS *****
38   messageIndex = sim808.isSMSunread();
39   //***** At least, there is one UNREAD SMS *****
40   if (messageIndex > 0){
41     Serial.print("messageIndex: ");
42     Serial.println(messageIndex);
43     sim808.readSMS(messageIndex, message, MESSAGE_LENGTH, phone, datetime);
44     //*****In order not to full SIM Memory, is better to delete it*****
45     sim808.deleteSMS(messageIndex);
46     Serial.print("From number: ");
47     Serial.println(phone);
48     Serial.print("Datetime: ");


```

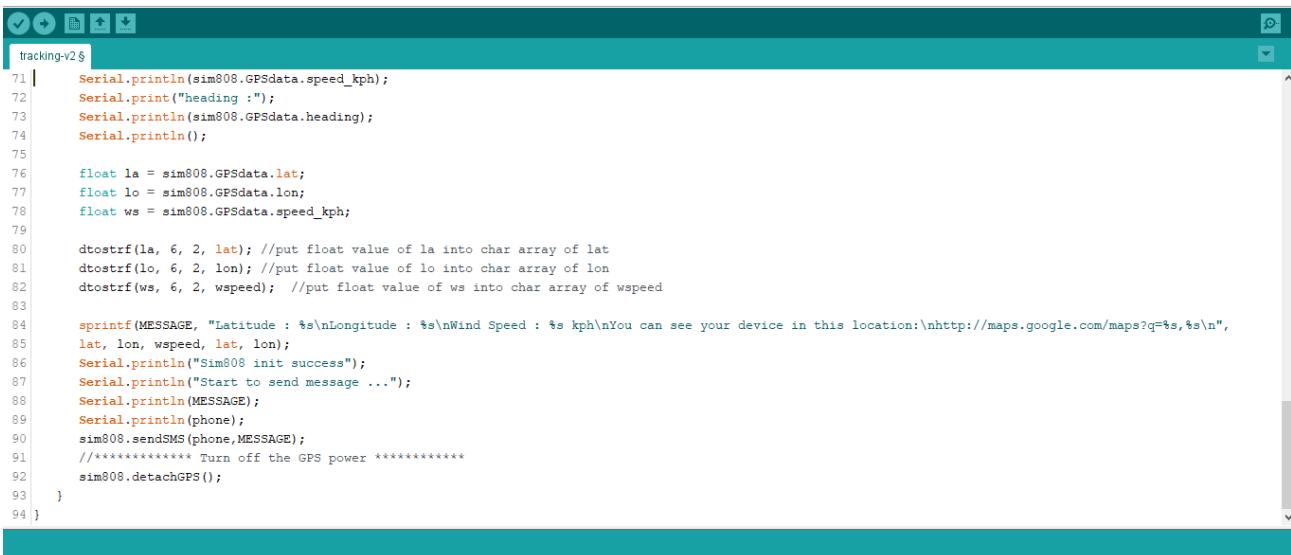


```

tracking-v2 $

48     Serial.print("Datetime: ");
49     Serial.println(datetime);
50     Serial.print("Received Message: ");
51     Serial.println(message);
52     while(!sim808.getGPS()){}
53     Serial.print(sim808.GPSdata.year);
54     Serial.print("/");
55     Serial.print(sim808.GPSdata.month);
56     Serial.print("/");
57     Serial.print(sim808.GPSdata.day);
58     Serial.print(" ");
59     Serial.print(sim808.GPSdata.hour);
60     Serial.print(":");
61     Serial.print(sim808.GPSdata.minute);
62     Serial.print(":");
63     Serial.print(sim808.GPSdata.second);
64     Serial.print(":");
65     Serial.println(sim808.GPSdata.centisecond);
66     Serial.print("latitude :");
67     Serial.println(sim808.GPSdata.lat);
68     Serial.print("longitude :");
69     Serial.println(sim808.GPSdata.lon);
70     Serial.print("speed_kph :");
71     Serial.println(sim808.GPSdata.speed_kph);

```



```

tracking-v2 $

71     Serial.println(sim808.GPSdata.speed_kph);
72     Serial.print("heading :");
73     Serial.println(sim808.GPSdata.heading);
74     Serial.println();
75
76     float la = sim808.GPSdata.lat;
77     float lo = sim808.GPSdata.lon;
78     float ws = sim808.GPSdata.speed_kph;
79
80     dtostrf(la, 6, 2, lat); //put float value of la into char array of lat
81     dtostrf(lo, 6, 2, lon); //put float value of lo into char array of lon
82     dtostrf(ws, 6, 2, wspeed); //put float value of ws into char array of wspeed
83
84     sprintf(MESSAGE, "Latitude : %s\nLongitude : %s\nWind Speed : %s kph\nYou can see your device in this location:\nhttp://maps.google.com/maps?q=%s,%s\n",
85     lat, lon, wspeed, lat, lon);
86     Serial.println("Sim808 init success");
87     Serial.println("Start to send message ...");
88     Serial.println(MESSAGE);
89     Serial.println(phone);
90     sim808.sendSMS(phone,MESSAGE);
91     //***** Turn off the GPS power *****
92     sim808.detachGPS();
93 }
94 }

```