

پروژه دزدگیر خودرو به همراه ردیاب GPS و نرم افزار ویندوز



دانشگاه اصفهان

دانشگاه اصفهان - دانشکده فنی و مهندسی - گروه مهندسی برق

کارگاه سیستم های نهفته

استاد مسیح محمد شفیعی

اعضا:

امیر زینالی

سید حسین مرتضوی

پویش کوره داوودی

ترم بهمن ۱۴۰۳ - تیر ۱۴۰۴

## فهرست مطالب:

فهرست مطالب:	۲
چکیده:	۳
مقدمه:	۴
۱ - طراحی سیستم:	۵
ماژول gsm sim800L:	۶
ارتباط http برای ارتباط با سرور از طریق gsm:	۷
ماژول رهیاب gps ublox neo 6m:	۷
BLE (Bluetooth Low Energy):	۸
فرستنده و گیرنده RF:	۸
سایر قطعات بکار رفته:	۹
جدول قطعات به کار رفته:	۹
۲- طراحی نرم افزار:	۱۰
پنل نرم افزار:	۱۰
تنظیم محدوده امن برای وسیله:	۱۱
قضیه هاورسین (فرمول مسافت روی کره):	۱۳
فلوچارت کارکرد نرم افزار:	۱۴
۳ - تحلیل و نتیجه گیری:	۱۴
۴ - پیوست:	۱۵

## چکیده:

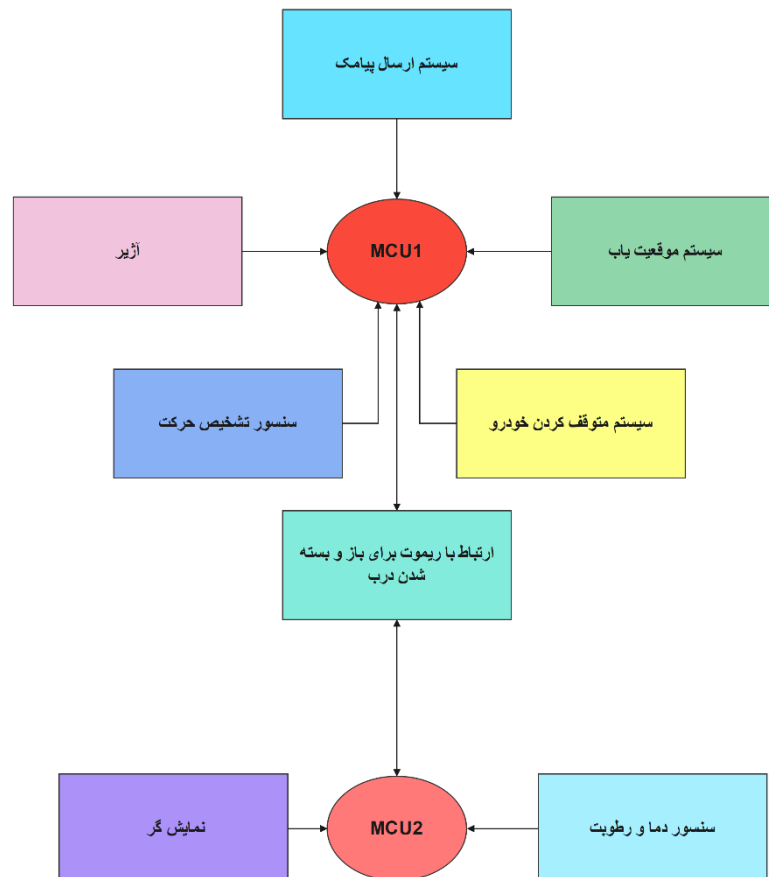
در این پروژه که پروژه پایانی درس کارگاه سیستم های نهفته است، تلاش ما بر این بود که سیستمی طراحی کنیم که قابلیت های هوشمند تری از یک دزدگیر معمولی داشته باشد و همچنین توانایی های اعضای گروه را به چالش بکشد. ما با استفاده از مبانی که در کلاس یادگرفتیم، بلد بودیم و تحقیقاتی که کردیم، این پروژه را به انجام رساندیم. قابلیت های متمایز کننده پروژه عبارت است از؛ قابلیت باز کردن یا قفل کردن خودکار درب، رهیابی ماهواره ایی خودرو، اعلان خطر با کمک پیامک یا تماس، کنترل مکان خودرو از راه دور به وسیله نرم افزار و تعیین محدوده ایمنی برای آن.

## مقدمه:

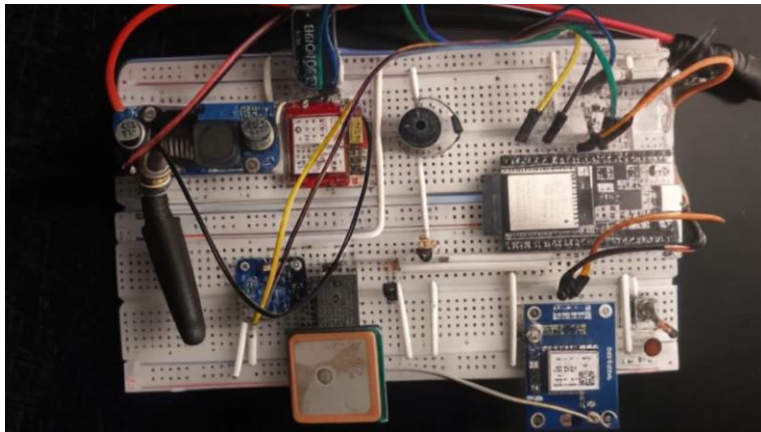
پروژه گروه در کارگاه سیستم های نهفته ، دزدگیر خودرو و رهیاب ماهواره ایی هوشمند، در واقع به این علت انتخاب شد که هم در دانشگاه به نوعی بدیع بود و هم به اندازه کافی چالش برانگیز. هدف نهایی در این پروژه ساخت بردی برای کنترل کلی ماشین و به ویژه رهیابی آن به وسیله یک نرم افزار بود. ساختار این گزارش به این صورت است که در بخش طراحی سیستم، بلوک دیاگرام، تصاویر، قطعات و ماژول های الکترونیکی قرار دارند و توضیحات در رابطه با انتخاب قطعات و ماژول ها و چگونگی کار کرد سیستم در این بخش آورده شده است. در بخش طراحی نرم افزار، در ارتباط با نرم افزار ویندوزی که برای این پروژه طراحی شد، تصاویر و توضیحات مربوطه قرار دارد و در بخش نتیجه گیری، تجربیاتی که این پروژه به گروه اضافه کرد و همچنین نتایج و زمینه های توسعه بیشتر پروژه آورده شده است. در بخش پیوست نیز کد بخش الکترونیکی و نرم افزار پروژه قرار داده شده است.

## ۱ - طراحی سیستم:

بلاک دیاگرام کلی سیستم به صورت زیر است و کل پروژه در واقع از ۲ برد تشکیل شده که تصاویر آنها هم در تصویر شماره ۲ و ۳ قابل مشاهده است.



تصویر ۱ - بلوک دیاگرام کلی پروژه

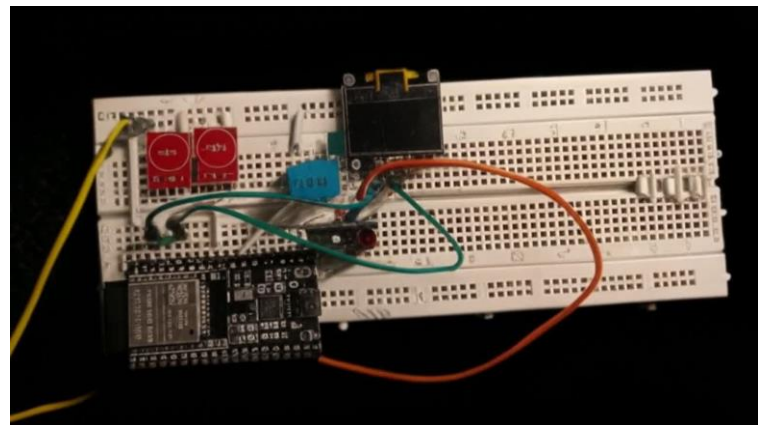


نصویر 3

برد اصلی که برای ماشین  
طراحی شده و شامل gps،  
RF receiver و gsm

نصویر 2

برد ریموت برای کنترل باز و بسته  
بودن قفل و نمایش اطلاعات سنسور  
دما و رطوبت، شمال سنسور تاج برای  
قفل و باز کردن.



### ماژول sim800L: gsm

به منظور ارتباط گرفتن با صاحب خودرو و ارسال پیام هایی از قبیل جابه جا شدن یا باز شدن درب یا تحرکات در ماشین به صاحب خودرو از ماژول sim800l استفاده شده که در این پروژه کافی به نظر می رسيد. از آنجا که در این پروژه ما از امکانات دیگر آن از قبیل میکروفن و اسپیکر آن استفاده نمی کنیم پس سخت افزار های مورد نیاز آن دو عدد خازن ۲۲۰۰uf و ۱۰۰nf برای تغذیه ی پایدار و کم نویز یک عدد آنتن GSM و دو عدد مقاومت ۵.۶k و ۱۰k برای RX خواهند بود. دلیل استفاده از این مقاومت ها ایجاد یک تقسیم مقاومتی است چرا که دریافت کننده GSM تحمل ۳.۳ ولت را ندارد. همچنین تغذیه ی GSM به طور کلی از بقیه ی مدار جدا است و با ماژول lm2596 تهیه می شود چرا که GSM بازه ولتاژی بین ۴.۴ و ۳.۷ را نیاز دارد و در غیر این سورت یا می سوزد و یا که روشن نمی شود. بنابراین به دلیل این که ماژول GSM و MCU بتوانند با مرجع ولتاژی درستی با یکدیگر ارتباط بگیرند لازم است تا زمین هردو مدار به یکدیگر متصل باشد تا بتوان بین این دو ارتباط برقرار کرد. در پیوست که کد esp آورده شده است در ادامه توضیحاتی در مورد توابع مربوط به gsm آمده است. bool sendATCommand() تابعی عمومی برای ارسال دستورات AT به ماژول SIM800L و بررسی پاسخ ها. bool sendSMS() با استفاده از sendATCommand() یک پیامک به شماره صاحب خودرو ارسال می کند. پیام ها مربوط به هشدار نفوذ یا حرکت هستند. makeCall() تماس صوتی با صاحب خودرو در صورت تشخیص حرکت خودرو در حالت هشدار را برقرار می کند.

## ارتباط http برای ارتباط با سرور از طریق gsm:

یک برد کاملاً مستقل که به عنوان مثال در داخل یک خودرو قرار می‌گیرد نمی‌تواند از ارتباط wifi برای ارسال اطلاعات خود به سرور استفاده کند، مگر آنکه اینترنت محلی سراسری برقرار باشد که متأسفانه این امکان در ایران مهیا نیست. بنابراین باید از طریق سیم کارت درون gsm ارتباط مذکور برقرار شود. ماژول gsm توانایی ارسال اطلاعات از طریق داده سیم کارت به وسیله http post را دارد برای اینکار تنها لازم است قبل از ارسال این اطلاعات AT COMMAND های مناسب برای gsm ارسال شود که به خوبی در بلوک تابع `sendThingSpeakPost()` در پیوست این کامند ها مشخص اند. به عنوان مثال ("`AT+SAPBR=3,1,\"Contype\",\"GPRS\"`") این کامند نوع کانکشن را روی gprs میگذارد یا کامند ("`AT+HTTPINIT`")، ارتباط http را شروع می‌کند. مسئله ایی که حائز اهمیت است برای کارکرد درست کد حتماً باید این ارتباط باز شده را بعد از ارسال اطلاعات بست چراکه در بسیاری از موارد شرکت های اوپراتور در صورتی که فعالیتی از این نوع ارتباطات نبینند آنها را خود به خود غیر فعال میکنند و برای ارسال های بعدی به مشکل بر میخوریم. با اجرا این کامند کل فرایند بسته میشود ("`AT+SAPBR=0,1`").

در ارتباط با سرور باید گفت که این پروژه از سرور دانشجویی رایگان ولی با محدودیت thingspeak که از mathworks هست استفاده می‌کند و برای ارسال اطلاعات همراه اطلاعات با کد ارسال سرور (Write api key) و برای دریافت کد مربوط (Get api key) را برای سرور بفرستیم و همچنین برای مشخص کردن url سرور برای gsm باید از این کامند ("`AT+HTTPPARA=\"URL\", \"http://api.thingspeak.com/update\"`") استفاده کنیم که "`http://api.thingspeak.com/update`" آدرس سرور می‌باشد.

## ماژول رهیاب gps ublox neo 6m:

برای ارتباط گرفتن با ماهواره های موقعیت یاب نیاز به یک چیپ و آنتن مخصوص این کار هست ما در پروژه در ماژول ublox neo 6m استفاده کردیم که هم چیپ و هم آنتن سرامیکی passive ایی برای برقراری این ارتباط دارد. علت انتخاب این ماژول قیمت مناسب تر آن برای انجام یک پروژه دانشجویی و رایج بودن آن در بازار بود. این ماژول از دقت خیلی خوبی برای تعیین موقعیت خودش برخوردار است و به گفته دیتاشیت شرکت سازنده آن ۲ تا ۲/۵ متر خطای مکان یابی آن است که برای پروژه ما این دقت بسیار هم عالی است. و نکته دیگری که در مورد این ماژول هست این است که ublox neo 6m صرفاً از ماهواره های موقعیت یاب آمریکایی GPS پشتیبانی می‌کند و برای ارتباط با آنها طراحی شده است، در این حال که ماژول دیگری از همین برند ublox neo m8m می‌تواند با چند سامانه موقعیت یاب ارتباط همزمان بگیرد به عنوان مثال GPS آمریکایی و GLONASS روسی، و در نتیجه میتواند با دقت بیشتری موقعیت یابی بکند.

از ارتباط UART برای اتصال ماژول به برد ESP32 استفاده شد. همانطور که در پیوست آورده شده RX ماژول به پین ۴ برد ESP32 و TX آن به پین ۵ آن متصل است و برای ارتباط از Baud rate ۹۶۰۰ هرتز استفاده می‌شود. بعد از برقرار ارتباط uart و تامین برق ۳.۳ ولتی مورد نیاز، ماژول هنگامی که با ماهواره ها ارتباط خود را برقرار کرد شروع به چشمک زدن می‌کند و از آن پس پیوسته اطلاعات مربوطه را می‌فرستد. این اطلاعات به فرمت NMEA است که استاندارد دریانوری آمریکا هست و

توسط تمامی سامانه های موقعیت یاب مورد اتفاق می باشد. برای ترجمه این اطلاعات از کتابخانه ایی تحت عنوان `tinygpsplus.h` در آردوینو استفاده می شود که مطابق بلوک تابع `gpsmodule()` در پیوست میتوان از آن استفاده کرد. `Lat` و `lng`، سرعت، دقت، تعداد ماهواره های در دسترس، ارتفاع از سطح زمین و بسیاری اطلاعات دیگر که `gps` به ما میدهد توسط این کتابخانه ترجمه میشوند و در نرم افزار که در ادامه توضیحات مربوط به این اطلاعات بیشتر آمده نمایش داده میشوند. در تابع ذکر شده نیز ما با استفاده از سیم کارت درون `gsm` با سرور `thingspeak by matlab` ارتباط میگیریم و سپس این اطلاعات در نرم افزار از سرور گرفته می شوند و به کابر هر ۲۰ ثانیه یک بار نمایش داده میشوند در این باره در بخش "ارتباط `http` برای ارتباط با سرور با `gsm`" مفصل تر توضیح داده شده است.

### BLE (Bluetooth Low Energy)

برای باز شدن درب خودرو به صورت خود به خود از BLE استفاده شده که خود ماژول `ESP32` دارای آن است چرا که رنج محدود تر و انرژی کم تری نسبت به بلوتوث کلاسیک دارد. در رابطه با کدهای مربوطه در پیوست آمده است؛ `class MyAdvertisedDeviceCallbacks` که برای تشخیص حضور دستگاه BLE صاحب خودرو (بر اساس `RSSI`). اگر قدرت سیگنال کافی باشد، فرض می شود مالک نزدیک است. بلوک تابع `handleBLELocking()` برای آن است که اگر دستگاه BLE نزدیک باشد، قفل درب باز می شود و اگر دور شود و چند ثانیه ناپدید بماند، درب قفل می شود. تمام متغیرهای هشدار نیز ریست می شوند. `handleBLELocking()` این تابع وضعیت حضور BLE را مدیریت می کند: اگر BLE شناسایی شود (یعنی صاحب خودرو نزدیک است) و در حال حاضر درب قفل باشد، دستور باز کردن درب اجرا می شود. اگر BLE ناپدید شود برای چند ثانیه، و در حال حاضر درب باز باشد، درب قفل می شود. همچنین هشدارها و آژیر خاموش می شوند تا ورود مالک به خودرو باعث فعال ماندن هشدار نشود. در کل برای انجام این مراحل و کل این ارتباطات از کتابخانه `BLEDevice.h` استفاده می شود.

### فرستنده و گیرنده RF

برای باز شدن درب با استفاده از RF به شکل معمول و رایج از ماژول های فرستنده و گیرنده `ASK` استفاده شده که هر دو با ۵ ولت تغذیه می شوند و نیاز به سخت افزار اضافه تری ندارند. `uint8_t generateToken()` این بلوک تابعی است که در کد پیوست یک توکن امنیتی ساده تولید می کند تا ارتباط RF با ریموت امن تر شود. پیام و کلید مخفی به صورت `XOR` ترکیب می شوند تا یک عدد هش به دست آید. همین تابع در رسیور نیز استفاده شده تا توکن فرستاده شده درست و صحیح باشد. در بخش ریموت RF بلوک تابع `sendSecureCommand()` وظیفه ارسال فرمان RF را بر عهده دارد. ابتدا با استفاده از `generateToken` یک توکن امنیتی تولید کرده، سپس بسته ای به صورت `cmd:token` ساخته و از طریق `RH_ASK` ماژول `RF ASK 315MHz` ارسال می کند. در نهایت، مقدار فرمان ارسال شده در متغیر `lastAction` ذخیره می شود تا برای نمایش روی `OLED` مورد استفاده قرار گیرد.



### سایر قطعات بکار رفته:

برای شبیه سازی باز و بسته شدن درب خودرو از یک push button استفاده شده که به منظور debounce کردن آن از یک خازن ۱۰۰nf استفاده شده و یک سمت آن با مقاومت پول آپ شده تا کلید فلوت نباشد. همچنین به منظور شبیه سازی آژیر از یک بازر پسیو استفاده شده چرا که فرکانس های مختلف را می توان با آن تولید کرد. ولی از آنجا که هرچند کم ولی دارای بار سلفی است نیاز به یک عدد دیود شاتکی بوده تا اثر بار سلفی را خنثی کند و مانع آسیب به مدار شود. بلوک توابع beepOnce() و beepDouble() مربوط به تولید صدای بازر هستند.

از یک عدد رله نیز برای شبیه سازی رله قطع کن خودرو استفاده شده این رله استارت خودرو را قطع می کند و پس از متوقف شدن خودرو در صورت فعال بودن رله خودرو دیگر روشن نخواهد شد. این یک آپشن امنیتی است که با ارسال پیامک فعال می شود. برای فعال کردن رله به طور مستقیم از سمت میکرو نمی توان اقدام کرد بلکه باید از یک ترانزیستور c945 استفاده کرد که بیس آن از طریق یک مقاومت ۴.۷k تحریک می شود و یکی از پایه های بوبین رله را به ۵ ولت متصل می کند و از آنجا که پایه ی دیگر بوبین به زمین متصل است رله تحریک می شود. همانند بازر رله نیز نیاز به یک دیود در دوسر پایه های بوبین دارد.

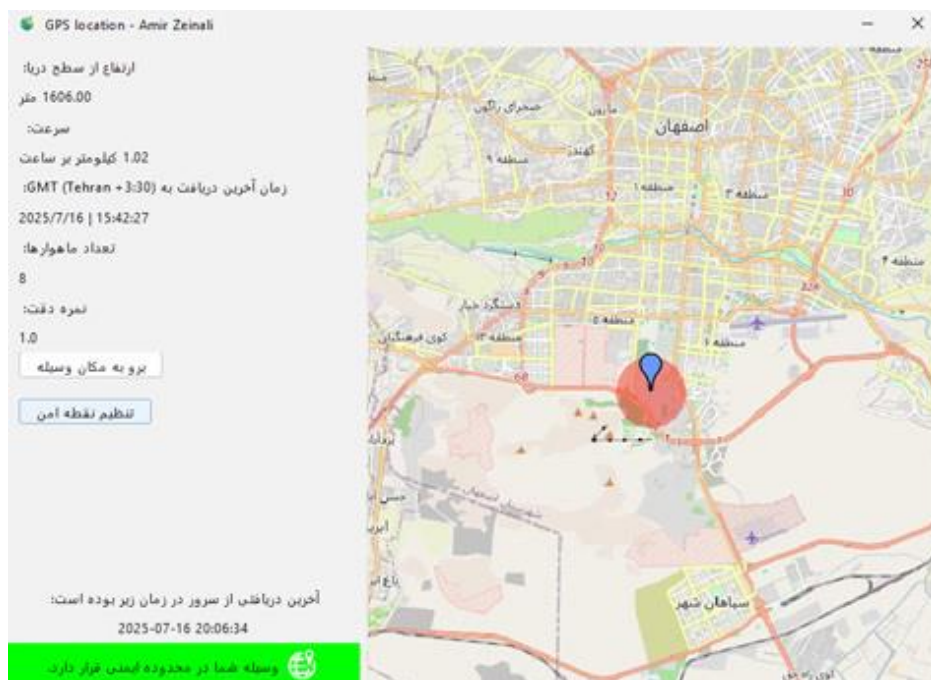
از امکانات دیگر این پروژه می توان به نمایشگر ریموت اشاره کرد که یک عدد OLED 0.96 inch است که به سادگی با I2C کار می کند. سنسورهایی نیز در پروژه استفاده شده اند که ما از سنسور دما dht11 و mpu6050 استفاده کردیم که نحوه کار با هر کدام را پیش از این در طول ترم قرا گرفتیم.

برای تشخیص وجود انسان یا موجود زنده در داخل کابین نیز از سنسور PIR که از نور مادون قرمز استفاده می کند استفاده شده. که تغذیه ی آن ۵ ولت است.

### جدول قطعات به کار رفته:

قطعات		
Ublox neo 6m	GSM SIM800L	ESP32WROOM-32D
LM2596 DC-DC CONV.	315MHz RF Transmitter	315MHz RF Reciever
DHT11	OLED 0.96"	MPU6050
2x TTP223	5V Relay	Passive Buzzer
.....	Resistors	Push button

## ۲- طراحی نرم افزار:



برای استفاده حداکثری از قابلیت های برد و همچنین کامل کردن پروژه یک نرم افزار ویندوزی برای نمایش موقعیت دستگاه در این مورد خودرو ساخته شد. در حالت عادی وب سایت هایی موجود هست که میتوان lat و lng را مستقیماً اطلاعات خام gps را به آنها داد و آنها مکان شما را روی نقشه نمایش نمی دهند اینکار

شاید برای یک آزمودن مازول مناسب باشد ولی برای یک پروژه اینطور نیست. این نرم افزار با زبان برنامه نویسی java و به صورت دقیق تری java swing و java fx طراحی شده برای کامپایل و گرفتن ارور ها نیز از ide رایج بین برنامه نویسان جاوا یعنی Eclipse ide استفاده شده است.

### پنل نرم افزار:

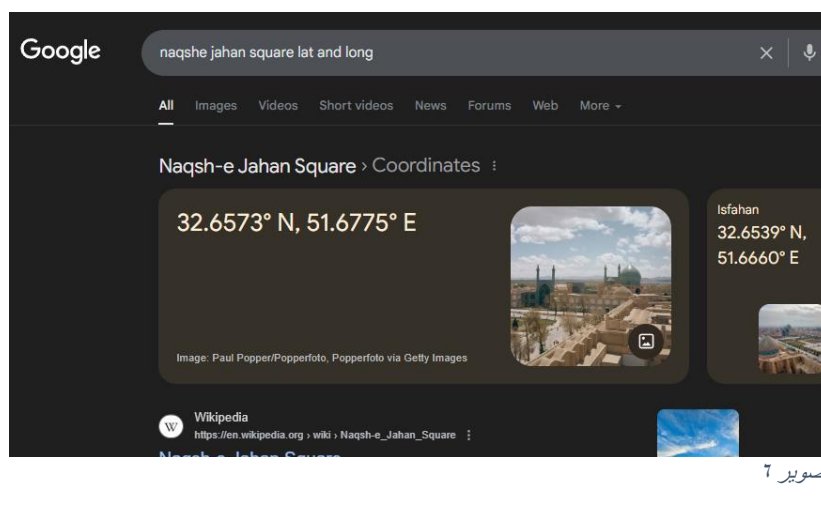
پنل نرم افزار همانطور که از تصویر ۴ مشخص است سمت راست نرم افزار قرار دارد. در این پنل ۵ مورد از ۷ تا از مهمترین اطلاعاتی که مازول gps به ما میدهد آمده است. ارتفاع از سطح دریا، سرعت به کیلومتر بر ساعت، آخرین فعالیت ارسالی مازول به وقت utc گرینویچ تعداد ماهواره های در دید مازول، و دقت و قدرت ارتباط. در ارتباط با زمان، تایم دریافت اطلاعات از سرور یعنی آخرین درخواست نرم افزار برای دریافت اطلاعات از سرور در پایین صفحه به وقت محلی نمایش داده شده که اگر مازول همواره هر ۲۰ ثانیه به درستی در حال فعالیت باشد این دو زمان کاملاً با یکدیگر برابر خواهند بود. و در مورد دقت هم باید گفت که با توجه به ساختار NMEA اطلاعات هرچه این مقدار به ۱ و صفر نزدیکتر باشد یعنی دقت بالاتر است و هر چه به ۹۹ نزدیکتر باشد یعنی دقت کمتر و توانایی اتصال با ماهواره ها به مشکل برخورده است به عنوان مثال مازول در محیط بسیار بسته ای قرار دارد، که قبلاً نیز اشاره شد که این مازول برای کارهای خیلی دقیقتر در مناطق شلوغ و سر بسته مناسب نیست و نیاز به تهیه مازول ها یا قطعات با قیمت های بالا تری هست. همچنین در داخل پنل یک نوار طراحی شده است که هم اکنون سبز رنگ است در تصویر بالا، و نشان دهند آن است که آیا وسیله در محدوده ایمنی تعیین شده قرار دارد یا خیر (دایره قرمز رنگ در نقشه) که جلوتر به تنظیم آن اشاره میشود. همچنین دو دکمه نیز در پنل تعبیه شده است که با کلیک بر روی (برو به مکان وسیله) در صورتی که در نقشه جا به جا شده باشیم میتوان به مکان وسیله

برگشت یا با زوم بیشتری وسیله را نگاه کرد. هرچند هر ۲۰ ثانیه یکبار موقعیت نقشه به مکان اولیه و زوم پیش فرض بر می گردد و برای زوم بیشتر مانند باقی نقشه ها از موس باید استفاده کرد. با کلیک بر روی دکمه (تنظیم نقطه امن) صفحه زیر باز می شود، که از این طریق میتوان محدوده ایمنی را تعیین کرد:

تصویر ۵

### تنظیم محدوده امن برای وسیله:

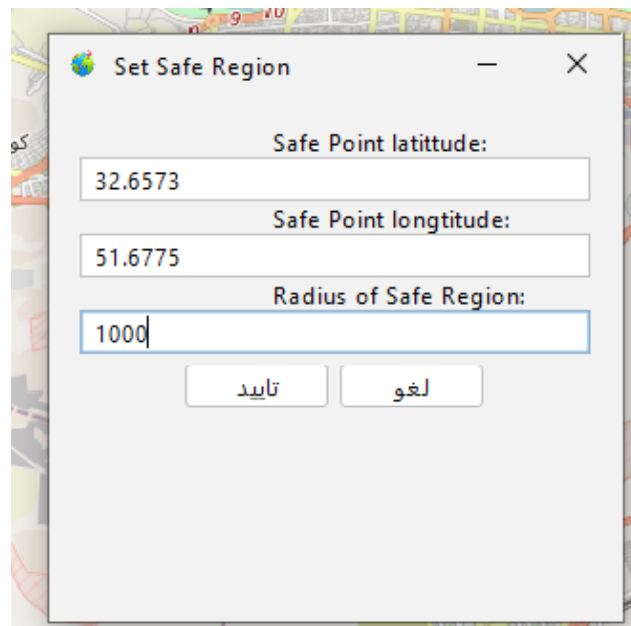
برای انجام این کار بعد از انتخاب (تنظیم نقطه امن) تصویر بالا باز میشود، شما میتوانید یک مکان را به عنوان مثال پارکینگ خودرو را به عنوان مکانی امن انتخاب کنید، سپس نزد خود به این پرسش پاسخ دهید که تا چه شعاعی برای آن وسیله ایمن محسوب میشود به عنوان مثال شب است و باید ماشین نهایتاً در محله باشد. پس از پاسخ به این سوالات lat مکان خود را در فیلد اول، lng مکان خود را در فیلد دوم، شعاع این محدوده ایمنی را در فیلد سوم به متر وارد کنید. سپس روی گزینه (تایید) کلیک کنید و پنجره را ببندید، نرم افزار در دور بعدی بررسی خود و گرفتن اطلاعات از سرور موقعیت وسیله شما را بررسی میکند و به شما نتیجه را از طریق نوار پتل راست برنامه که پیشتر به آن اشاره شد نمایش میدهد.



تصویر ۶

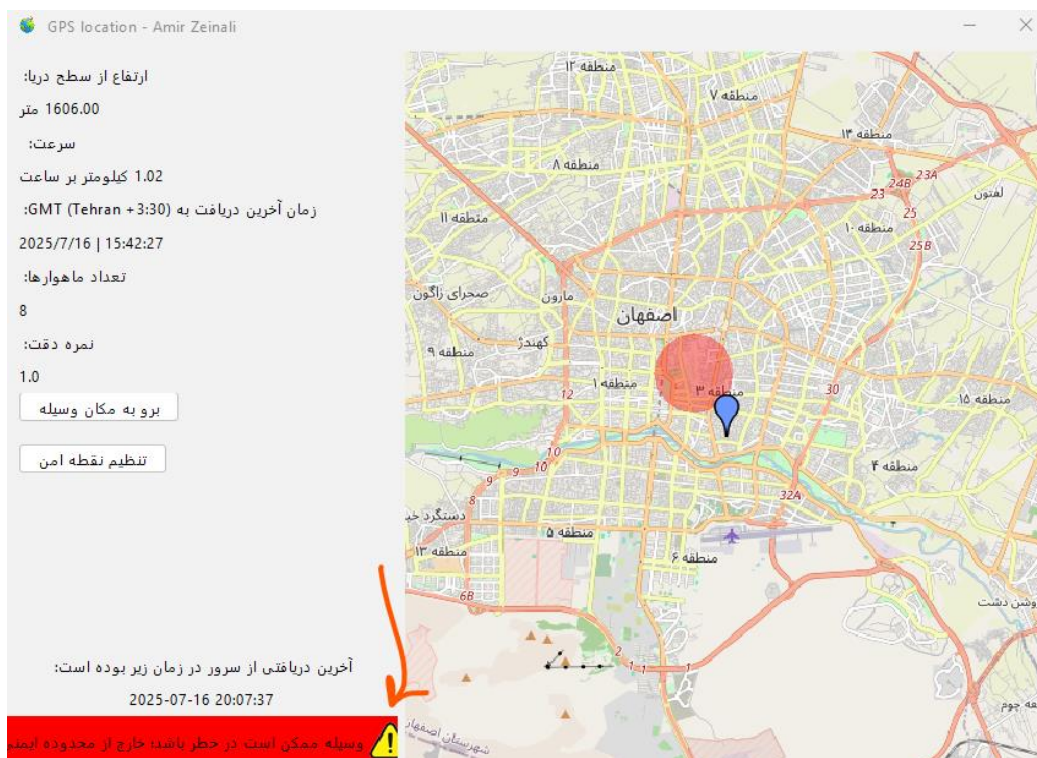
برای مثال ما موقعیت میدان نقش جهان اصفهان را در اینترنت سرچ میکنیم و لت و لانگ جغرافیایی آن را بر میداریم:

Lat (طول جغرافیایی) و Lng (عرض جغرافیایی) را کپی کرده و شعاع ۱۰۰۰ متری (۱ کیلومتر) را برای محدوده ایمنی مطابق شکل انتخاب میکنیم:



نصویر ۷

با کلیک بر روی گزینه تایید تنظیمات ما ذخیره میشود و برنامه بعد از گذشت حداکثر ۲۰ ثانیه در دور بعدی بررسی و گرفتن



نصویر ۸

### قضیه هاورسین (فرمول مسافت روی کره):

برای محاسبه فاصله مکان از مرکز نقطه امن از قضیه هاورسین (فرمول مسافت روی کره) استفاده شده است به آن علت که نقشه ها خود ساختار کروی زمین را در بطن هویتی خود حفظ می کنند با وجود آن که نقشه ها در صفحه ها به صورت تخت نمایش داده میشوند:

$$\varphi = \text{latitude}, \lambda = \text{longitude}$$

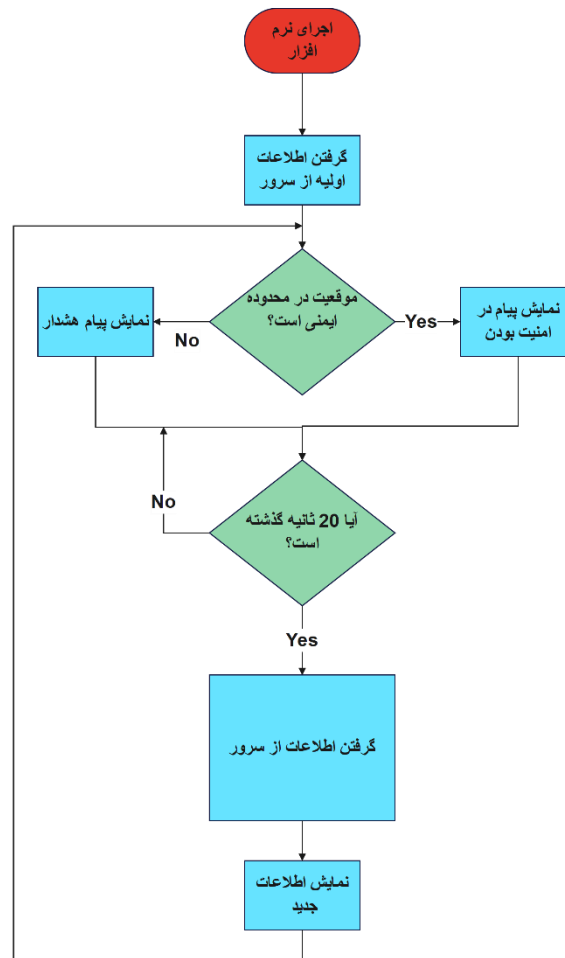
$$\Delta\varphi = \varphi_2 - \varphi_1 \quad \Delta\lambda = \lambda_2 - \lambda_1$$

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1) \cdot \cos(\varphi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)$$

$$c = 2 \cdot \arctan2(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

## فلوچارت کارکرد نرم افزار:



## ۳- تحلیل و نتیجه گیری:

پروژه با موفقیت اجرا شد و ماژول های GPS، GSM، RF و ... به خوبی وظایف مورد انتظار را انجام دادند و اطلاعات نیز به خوبی از سرور گرفته یا به آن فرستاد شد. اما مسائلی که میتواند به توسعه پروژه کمک کند استفاده یک ماژول قدرتمند تر برای GPS یا آنتن اکتیو که دقت بالاتری به ما می دهد. به علاوه ماژول GSM قدرتمند تری می تواند مشکلاتی اعم از اتصال اینترنت و آنتن دهی را مرتفع کند. چراکه ماژول SIM800L ارتباطات قدیمی تری را فراهم می کند مثل 2G که رفته رفته در بسیاری از کشور ها در حال منسوخ شدن است. نرم افزار نیز میتواند در مورد تعیین نقطه امن رابط کاربری مناسب تری ارائه دهد و نیز هنگامی که خودرو خارج از محدود قرار می گرفت جدای از اعلان در خود نرم افزار به فرد نیز پیامکی ارسال کند از ارتباط گرفتن از طریق سرور با ESP.

#### ۴ - پیوست:

- فایل کدهای سخت افزار در محیط آردوینو در پیوست همین فایل قرار داده شده است.

[carAlarm\\_and\\_gpsTraking\\_project.ino | open file](#)

- فایل rar حاوی source های بخش هایی از نرم افزار به عنوان نمونه در پیوست این فایل قرار داده شده است:

[smartGPS\\_tracking\\_app.rar | open file](#)