



گزارش پایانی آزمایشگاه سخت افزار

گروه سه

اعضا:

بهار خدابخشیان ۹۷۱۰۵۹۰۶

محمدامین شریفی ۹۷۱۱۰۰۵۸

درنا دهقانی ۹۷۱۰۵۹۳۹

## فهرست

۱. مقدمه .....	۳
۲. نحوه انجام پروژه .....	۳
۳. مازول ها .....	۳
۳,۱. Raspberry Pi 3.....	3
۳,۲. LED .....	4
۳,۳. سنسور دما .....	4
۳,۴. برد یورد .....	4
۳,۵. سنسور مکان .....	5
۴. معماری سیستم .....	۵
۵. نحوه ارتباط سنسورها .....	۶
۵,۱. NEO-7M-C.....	6
۵,۲. RW1820 .....	6
۵,۳. LED .....	7
۶. گرفتن دما .....	۸
۷. گرفتن مختصات GPS .....	۹
۸. تشخیص جهت تابش آفتاب به کمک مختصات .....	۱۴
۹. محاسبه جهت حرکت جسم (خودرو) .....	۱۴
۱۰. صحت سنجی سنسور GPS .....	۱۵
۱۱. روشن و خاموش کردن خنک کننده ها .....	۱۶
۱۲. بخش اصلی پروژه .....	۱۸

## ۱. مقدمه

هدف از این پروژه، طراحی سیستمی برای متعادل سازی دمای خودرو می باشد، به طوریکه به مصرف انرژی به شکلی بهینه کمک کند. از آنجایی که در یک خودرو که در هوای آفتابی در حال حرکت است همواره یک سمت از آن در معرض نور خورشید قرار دارد، این بخش گرم تر از سایر بخش ها می باشد. اگر برای خنک کردن خودرو، خنک کننده ی تمامی قسمت ها را روشن کنیم، این کار موجب مصرف بی رویه انرژی در خنک کردن بخش هایی از خودرو می شود که در معرض نور مستقیم نیستند. برای جلوگیری از این مشکل، به کمک GPS و موقعیت خودرو می توانیم وضعیت قرارگیری آن نسبت به خورشید را بیابیم تا متوجه شویم کدام سمت آن رو به خورشید است. سپس به کمک حسگرهای دمای بخش های مختلف، به مقایسه آنها می پردازیم و در صورت تفاوت دما، خنک کننده ی آن بخش را روشن می کنیم تا دمای خودرو به تعادل برسد. نهایتاً پس از تعادل کامل دمای خودرو، خنک کننده ی خودرو خاموش می گردد. لازم به ذکر است که بررسی دمای بخش های مختلف خودرو باید با هر بار تغییر وضعیت آن نسبت به خورشید انجام گیرد. هم چنین، سیستم خنک کننده در قالب چراغ های LED پیاده سازی می شوند؛ یعنی اگر بخشی از خودرو نیاز به خنک کننده داشت، LED مربوط به آن روشن می گردد و اگر یک خنک کننده باید خاموش می شد، LED مربوط به آن خاموش می گردد.

## ۲. نحوه انجام پروژه

برای انجام این پروژه، ما از رزبری پای ۳ برای کنترل سیستم استفاده خواهیم کرد. برای بخش بندی خودرو آن را به ۴ قسمت راننده، شاگرد، و ۲ بخش سرنشینان عقب تقسیم می کنیم که در هر بخش یک حسگر دما قرار می گیرد. یک سنسور GPS نیز در خودرو قرار دارد و تمامی این سنسورها به رزبری پای متصل می گردند. جهت نمایش هشدارهای روشن و خاموش کردن چراغ های خنک کننده نیز از LED های متصل به رزبری پای استفاده می شود. برای نوشتن برنامه کنترلگر سیستم از زبان پایتون و کتابخانه های آن استفاده می کنیم.

## ۳. ماژول ها

### ۳.۱. Raspberry Pi 3

ما از این ماژول به عنوان پردازنده سیستم خود استفاده کردیم. این دستگاه با قابلیت نصب سیستم عامل، اجرای کد های پایتون و ارتباط با سنسور هایی از قبیل سنسور دما و GPS امکان ارتباط ساده بین اجزای سیستم را ارائه می دهد.



تصویر ۱ Raspberry Pi 3

### ۳,۲. LED

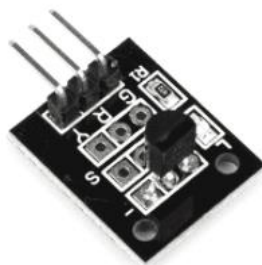
برای نمایش هشدارهای روشن و خاموش کردن خنک کننده در ۴ بخش خودرو، به ۴ عدد LED نیاز داریم.



تصویر ۲ LED

### ۳,۳. سنسور دما

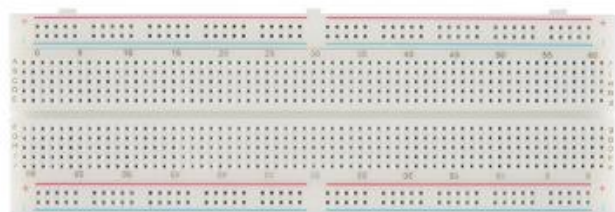
برای اندازه‌گیری دما در خودرو از سنسور دمای RW1820 استفاده خواهد شد. در این پروژه از ۴ عدد از این سنسور برای سنجش دمای ۴ ناحیه خودرو استفاده خواهیم کرد.



تصویر ۳ RW1820

### ۳,۴. بردبرد

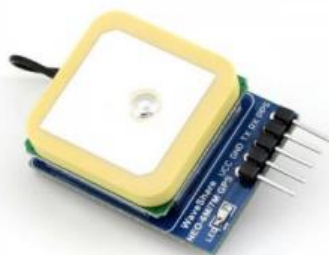
برای اتصال سنسورها و LED ها به رزبری پای به بردبرد احتیاج داریم.



تصویر ۴ بردبرد

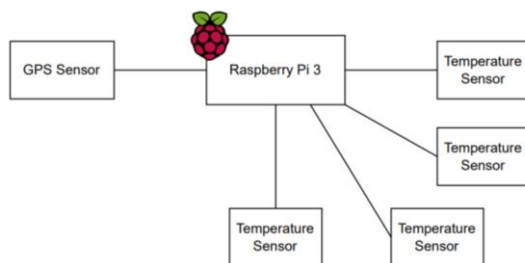
### ۳,۵. سنسور مکان

برای بدست آوردن مکان ماشین، از سنسور مکان NEO-7M-C استفاده خواهد شد. این سنسور با ریت ماکسیمم 5Hz این امکان را به ما می دهد تا موقعیت خودرو را بدست آوریم.



تصویر ۵ NEO-7M-C

### ۴. معماری سیستم

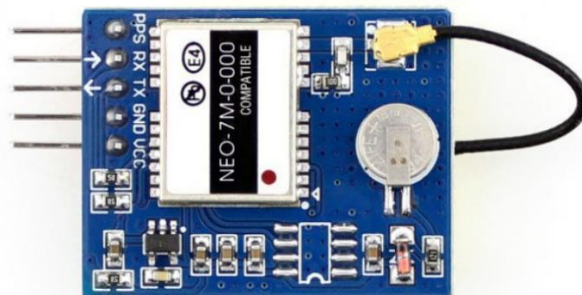


تصویر ۶ معماری سیستم

## ۵. نحوه ارتباط سنسورها

### ۵.۱. NEO-7M-C

این سنسور مکان یاب دارای ۴ پین است که ۲ تا از آنها برای GND و VCC هستند. دوتای دیگر با نام های RX و TX به ترتیب به پین های TXD و RXD در رزبری پای متصل خواهند شد.



تصویر ۷ پین های سنسور مکان یاب

پس از اتصال این پین ها باید config های لازم برای اتصال رزبری و سنسور را انجام دهیم. سپس باید دستورات زیر را برای UART انجام داد.

```
sudo raspi-config → Interfacing Options → Enable Serial
```

سپس برنامه های مورد نیاز برای ارتباط با سنسور را با دستور زیر نصب می کنیم.

```
sudo apt-get install gpsd gpsd-clients
```

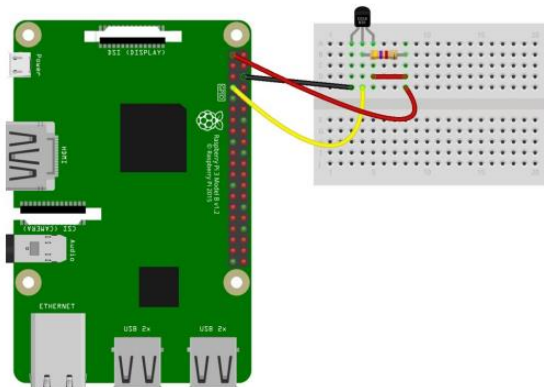
سپس با استفاده از دستور زیر می توان برنامه gpsd را اجرا کرد.

```
sudo gpsd /dev/ttyS0 -F /var/run/gpsd.sock
```

در آخر نیز با پکیج gps در پایتون می توان داده های لازم مکان را دریافت کرد. در هنگام پیاده سازی متوجه شدیم به دلیل مشکلات موجود در پیاده سازی کتابخانه، قابل استفاده نیست و نهایتاً تصمیم به استفاده از کتابخانه pynmea2 گرفتیم.

### ۵.۲. RW1820

مراحل اتصال این سنسور نیز مشابه مراحل سنسور قبلی است. این سنسور ۳ پین دارد که دوتا از آنها GND و VCC هستند. پین سوم که دیتا است باید به پین های GPIO در رزبری وصل شود با این تفاوت که نیاز به یک مقاومت نیز هست که با توجه به اینکه ما از ماژول این سنسور استفاده می کنیم، نیاز به مقاومت اضافی نیست.



fritzing

تصویر ۸ نحوه اتصال سنسور دما با رزبری پای

سپس مانند سنسور قبلی باید رابط با رزبری پای را تنظیم کنیم که با دستور زیر قابل انجام است.

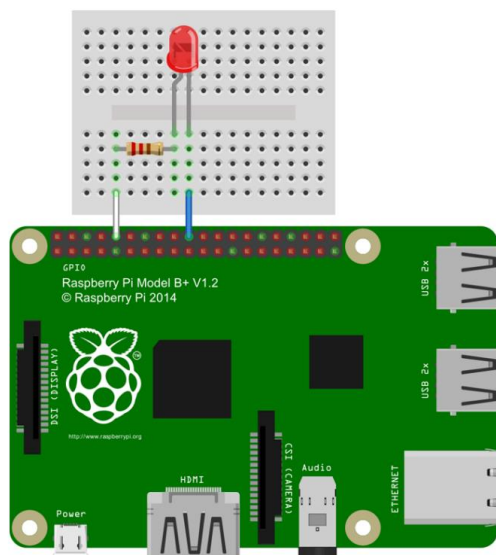
`sudo raspi-config` → Interfacing Options → Enable ۱-Wire

در انتها نیز با کتابخانه W1ThermSensor می توان اطلاعات دما را از سنسور دریافت کرد.

### ۵.۳. LED

در این پروژه قرار بر این است که از LED برای نشان دادن روشن بودن خنک کننده هر قسمت از ماشین استفاده کنیم.

LED را از طریق بردبورد به رزبری پای وصل می کنیم تا بتوانیم آن را با استفاده از برنامه پایتون و پکیج GPIO.RPi کنترل (روشن و خاموش) کنیم.



تصویر ۹ نحوه اتصال LED به رزبری پای

## ۶. گرفتن دما

برای این بخش از پروژه ماژول DS18B20 که شامل سنسور RW1820 می‌باشد را انتخاب کرده بودیم که به شکل ماژول به ما داده شد. در این ماژول خروجی‌های G, R و Y به ترتیب باید به پین‌های GND, VCC ۳٫۳ یا ۵ ولت و GPIO 4 (GPCLK0) متصل شوند. پین‌های [GPCLK](#) یا General Purpose Clock در رزبری پای پین‌هایی هستند که خروجی را با فرکانس ثابت و بدون نیاز به کنترل نرم‌افزاری ایجاد می‌کنند.

ابتدا می‌خواهیم نحوه نمایش دما توسط این سنسور را ببینیم. برای تنظیمات اولیه و ایجاد رابط یک طرفه بین ماژول و رزبری پای، پس از روشن کردن رزبری پای با زدن دستور `sudo nano /boot/config.txt`، در انتهای فایل باز شده عبارت `dtoverlay=w1-gpio` را اضافه می‌کنیم. سپس رزبری پای را **reboot** می‌کنیم. پس از روشن شدن مجدد و زدن دو دستور `sudo modprobe w1-gpio` و `sudo modprobe w1-therm` با دستور `cd /sys/bus/w1/devices` می‌توانیم لیست دایرکتوری‌های موجود در آن را مشاهده کنیم. دایرکتوری که اسم با فرمت `28-XXXXXXXXXXXX` دارند شامل اطلاعات دما هستند. با `cd` وارد آن شده و با دستور `cat w1_slave` فایل موجود در آن را باز می‌کنیم. با هر بار باز کردن این فایل، اطلاعات دمای گرفته شده در هر کلاک را مشاهده می‌کنیم که به شکل `t=xxxxxx` داده شده است و نشان می‌دهد دما `xxx . xx` درجه سانتیگراد است. از آنجایی که در این پروژه نیاز به ۴ ماژول دمای جداگانه داریم، باید برای دریافت اطلاعات از هر یک نیز جداگانه عمل کنیم. برای این کار کافیست هر ۱ یا ۲ ردیف از بردبرد را به یکی از پین‌های رزبری پای اختصاص دهیم و پین‌های هر یک از چهار ماژول دما به آن ردیف‌ها متصل کنیم.

پس از اتصال صحیح هر چهار ماژول دما، در پوشه `devices` که پیش‌تر تنها یک دایرکتوری که با فرمت اسم `28-XXXXXXXXXXXX` داشت، اکنون چهار دایرکتوری با این مشخصات قابل مشاهده است که فایل `w1_slave` در هر دایرکتوری اطلاعات دمای گرفته شده توسط آن ماژول را نمایش می‌دهد.

```
mmd@raspberrypi: ~/Desktop/project/hw_project-group-3/Code
Sensor 1 => 25.5 (C) - 77.9 (F)
Sensor 2 => 25.375 (C) - 77.675 (F)
Sensor 3 => 25.062 (C) - 77.11160000000001 (F)
Sensor 4 => 27.562 (C) - 81.61160000000001 (F)
Sensor 1 => 25.562 (C) - 78.0116 (F)
Sensor 2 => 25.375 (C) - 77.675 (F)
Sensor 3 => 25.0 (C) - 77.0 (F)
Sensor 4 => 26.562 (C) - 79.8116 (F)
Sensor 1 => 25.5 (C) - 77.9 (F)
Sensor 2 => 25.375 (C) - 77.675 (F)
Sensor 3 => 25.0 (C) - 77.0 (F)
Sensor 4 => 26.062 (C) - 78.9116 (F)
Sensor 1 => 25.5 (C) - 77.9 (F)
Sensor 2 => 25.375 (C) - 77.675 (F)
Sensor 3 => 25.0 (C) - 77.0 (F)
Sensor 4 => 25.75 (C) - 78.35 (F)
Sensor 1 => 25.562 (C) - 78.0116 (F)
Sensor 2 => 25.375 (C) - 77.675 (F)
Sensor 3 => 25.062 (C) - 77.11160000000001 (F)
Sensor 4 => 25.625 (C) - 78.125 (F)
Sensor 1 => 25.562 (C) - 78.0116 (F)
Sensor 2 => 25.375 (C) - 77.675 (F)
Sensor 3 => 25.062 (C) - 77.11160000000001 (F)
```

تصویر ۱۰ نمایش دمای چهار سنسور دما

در نهایت، برای استفاده از این دماها برای مراحل آتی از کد پایتون استفاده می‌کنیم که دما را از فایل‌های گفته شده می‌خواند. این کد در فایل `temp_sensor.py` قابل مشاهده است. اطلاعات ذکر شده در این بخش عمدتاً از [این لینک](#) گرفته شده است.



برای تشخیص اینکه هر سنسور دما بیانگر دمای چه بخشی از خودروست، به طور دستی به تغییر دمای محیط اطراف هر یک می‌پردازیم و سنسور دماها را به ترتیب ۰ (راننده)، ۱ (سرنشین عقب پشت راننده)، ۲ (شاگرد راننده) و ۳ (سرنشین عقب پشت شاگرد راننده) نامگذاری می‌کنیم.

## ۷. گرفتن مختصات GPS

برای دریافت مختصات خودرو از سنسور Neo-7m-C استفاده کردیم که ۴ پین VCC، GND، TX و RX آن به ترتیب به پین‌های ۳، ۳ یا ۵ ولت، GND، RX (GPIO 15) و TX (GPIO 14) متصل می‌شوند. برای برقراری اتصال بین سنسور و رزبری پای لازم است تنظیماتی انجام گردند. پس از روشن کردن رزبری پای با زدن دستور `sudo nano /boot/config.txt` در انتهای فایل باز شده عبارات

```
dtparam=spi=on
dtoverlay=pi3-disable-bt
core_freq=250
enable_uart=1
force_turbo=1
```

را اضافه کرده و ذخیره می‌کنیم. از آنجایی که سیستم عامل رزبری از UART به عنوان یک کنسول سریال استفاده می‌کند، برای خاموش کردن این عملکرد باید فایل `/boot/cmdline.txt` را تغییر دهیم. برای این کار ابتدا از نسخه اصلی آن با دستور `sudo cp /boot/cmdline.txt /boot/cmdline_backup.txt` یک فایل پشتیبان تهیه می‌کنیم. سپس با دستور `sudo nano /boot/cmdline.txt` آن را باز کرده و محتوای آن را با عبارت زیر جایگزین کرده و ذخیره می‌کنیم.

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2
rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait quiet
splash plymouth.ignore-serial-consoles
```

سپس رزبری پای را **reboot** می‌کنیم.

پس از این تنظیمات می‌توانیم کارکرد سنسور را مشاهده می‌کنیم. پس از اتصال سنسور به رزبری پای، یک LED قرمز رنگ روی آن روشن می‌شود که بیانگر روشن بودن سنسور است. پس از مدتی نسبتاً طولانی، یک LED سبز رنگ شروع به چشمک زدن می‌کند که بیانگر دریافت مختصات است. اگر این زمان از ۵ دقیقه بیشتر شد باید کارکرد آن در هوای آزاد یا نزدیک پنجره بررسی شود. پس از شروع چشمک زدن، با اجرای دستور `sudo cat /dev/ttyAMA0` اطلاعات مختصات دریافت شده توسط سنسور مطابق تصویر زیر نمایش داده می‌شود:

```

mmd@raspberrypi:/ $ sudo cat /dev/ttyAMA0
$GPRMC,,V,,,,,,,,,N*53

$GPVTG,,,,,,,,,N*30

$GPGGA,,,,,0,00,99.99,,,,,*48

$GPGSA,A,1,,,,,,,,,99.99,99.99,99.99*30

$GPGLL,,,,,V,N*64

$GPRMC,,V,,,,,,,,,N*53

$GPVTG,,,,,,,,,N*30

$GPGGA,,,,,0,00,99.99,,,,,*48

$GPGSA,A,1,,,,,,,,,99.99,99.99,99.99*30

$GPGLL,,,,,V,N*64

$GPRMC,,V,,,,,,,,,N*53

$GPVTG,,,,,,,,,N*30

$GPGGA,,,,,0,00,99.99,,,,,*48

$GPGSA,A,1,,,,,,,,,99.99,99.99,99.99*30

$GPGLL,,,,,V,N*64

$GPRMC,,V,,,,,,,,,N*53

$GPVTG,,,,,,,,,N*30

```

تصویر ۱۱ مختصات خام دریافتی از سنسور GPS

مطابق بخش قبل، برای استفاده از مختصات دریافتی سنسور باید از کد پایتون کمک بگیریم، اما پیش از آن نیاز به چند تنظیم دیگر داریم، چون رزبری پای به طور پیش فرض از پورت سریال برای کنسول لاگین استفاده می‌کند و ما به این پورت برای دریافت اطلاعات موقعیت مکانی نیاز داریم؛ پس باید کنسول لاگین را غیرفعال کنیم. در رزبری پای دو پورت سریال serial0 و serial1 وجود دارد که serial0 به GPIO 14 و GPIO 15 متصل است. برای مشاهده پورت‌های متصل به serial0 از دستور `ls -l /dev` استفاده می‌کنیم.

برای غیرفعال کردن کنسول از دستورات زیر استفاده می‌کنیم:

```

sudo systemctl stop serial-getty@ttyAMA0.service
sudo systemctl disable serial-getty@ttyAMA0.service

```

نهایتاً به کمک کد پایتون در فایل `gps.py` می‌توانیم اطلاعات موقعیت مکانی را خوانده و در بخش‌های بعدی از آن استفاده کنیم. اطلاعات ذکر شده در این بخش عمدتاً از [این لینک](#) گرفته شده است.

سنسور Neo-7m-c اطلاعات زیادی در خود دارد و یک نمونه از خروجی آن در تصویر زیر قابل مشاهده است، که البته به علت دوری از هوای آزاد خروجی دقیقی نیست:

```

mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ python gps.py
$GPRMC,105506.00,V,,,,,,,,181222,,,N*70
{'azimuth': -0.5518667376714196, 'altitude': 1.0859745784966284}
Latitude=0.0and Longitude=0.0
$GPRMC,105507.00,V,,,,,,,,181222,,,N*71
{'azimuth': -0.5517680835033895, 'altitude': 1.0860065621303725}
Latitude=0.0and Longitude=0.0
$GPRMC,105508.00,V,,,,,,,,181222,,,N*7E
{'azimuth': -0.551650420273802, 'altitude': 1.0860446960081322}
Latitude=0.0and Longitude=0.0
$GPRMC,105509.00,V,,,,,,,,181222,,,N*7F
{'azimuth': -0.5515328416997615, 'altitude': 1.0860827888870161}
Latitude=0.0and Longitude=0.0
$GPRMC,105510.00,V,,,,,,,,181222,,,N*77
{'azimuth': -0.5514152910748177, 'altitude': 1.0861208591628513}
Latitude=0.0and Longitude=0.0
$GPRMC,105511.00,V,,,,,,,,181222,,,N*76
{'azimuth': -0.5512976548651984, 'altitude': 1.0861589435997345}
Latitude=0.0and Longitude=0.0
$GPRMC,105512.00,V,,,,,,,,181222,,,N*75
{'azimuth': -0.551161113894121, 'altitude': 1.0862031314075655}
Latitude=0.0and Longitude=0.0
$GPRMC,105513.00,V,,,,,,,,181222,,,N*74
{'azimuth': -0.5510623858246105, 'altitude': 1.086235070714714}
Latitude=0.0and Longitude=0.0
$GPRMC,105514.00,V,,,,,,,,181222,,,N*73

```

تصویر ۱۲ خروجی سنسور و کد GPS

این اطلاعات به ترتیب عبارتند از:

اعتبارسنجی داده با XOR بین کاراکترهای * و \$ checksum	North/South	تاریخ	Navigation receiver status	زمان	نمایش شروع خط شامل اطلاعات
\$GPRMC	N	DDMMYY	V	HHMMSS.SSS	

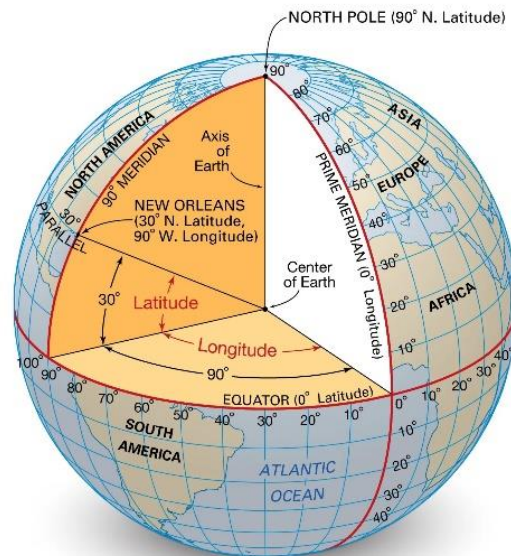
و همچنین latitude و longitude.

در ادامه با اجرای کد دریافت مکان خورشید، محل آن به کمک azimuth و altitude داده می‌شود. تعاریف این مفاهیم عبارتند از:

**Longitude:** طول جغرافیایی. بصورت زاویه‌ای بین منفی ۱۸۰ تا ۱۸۰ بیان می‌شود و فاصله یک نقطه روی زمین تا نصف‌النهار مبدا را نشان می‌دهد.

**Latitude:** عرض جغرافیایی. بصورت زاویه‌ای بین منفی ۹۰ تا ۹۰ بیان می‌شود و فاصله یک نقطه روی زمین تا خط استوا را نشان می‌دهد.

به کمک دو مقدار فوق، مختصات یک نقطه روی سطح زمین بدست می‌آید.



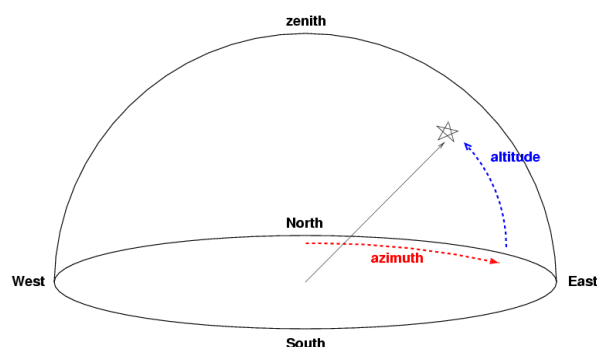
© Encyclopædia Britannica, Inc.

تصویر ۱۳ نمایش عرض و طول جغرافیایی

**Azimuth:** سمت. فاصله‌ی زاویه‌ای ستاره بر روی افق از شمال، که با حرکت در جهت شرق از ۰ تا ۳۶۰ درجه افزایش می‌یابد.

**Altitude:** ارتفاع. فاصله‌ی زاویه‌ای بر حسب درجه از افق که از ۰ تا ۹۰ درجه تغییر می‌کند.

به کمک دو مقدار فوق، محل یک نقطه در فضا بدست می‌آید.



تصویر ۱۴ نمایش دستگاه مختصات افقی

همانطور که در گزارش‌های پیشین گفته شد برای تشخیص جهت حرکت خودرو به مختصات آن در دو زمان نیاز است. برای این کار، پس از دریافت مختصات در یک لحظه، پس از ۵ ثانیه مجدداً مختصات جدیدی دریافت می‌گردد تا به کمک این دو مقدار جهت حرکت خودرو تعیین گردد. مقدار زمان ۵ ثانیه به این علت انتخاب شد که هم نیاز است خودرو در این مدت زمان مسافت قابل توجهی حرکت کرده باشد، و هم نباید آنقدر زیاد باشد که در این بین خودرو تغییر جهت داده باشد. همچنین اگر به هر علتی، از جمله ضعیف بودن سیگنال GPS، داده‌ی دوم پس از ۲۰ ثانیه از سنسور گرفته نشد، با هشدار Timeout مواجه می‌شویم و مجدداً تلاش به دریافت این دو مختصات می‌کنیم.

خروجی این سنسور در هوای آزاد در دانشگاه به شکل زیر است:

```
mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ nano gps.py
mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ python gps.py
$GPRMC,111217.00,A,3542.09769,N,05121.16434,E,1.789,,261222,,,A*77 $GPRMC,111222.00,A,3542.10127,N,05121.16710,E,3.129,35.02,261222,,,A*54
35.70162816666666 51.352785 35.70168783333333 51.352785 0.0 0.0
Latitude=35.70168783333333and Longitude=51.352785
Azimuth=3.002263500535479,Sun Angle=3.002263500535479
$GPRMC,111223.00,A,3542.10236,N,05121.16800,E,3.098,27.04,261222,,,A*56 $GPRMC,111229.00,A,3542.10737,N,05121.17271,E,3.582,27.90,261222,,,A*56
35.701706 51.3528785 35.7017895 51.3528785 0.0 0.0
Latitude=35.7017895and Longitude=51.3528785
Azimuth=3.000094510698154,Sun Angle=3.000094510698154
$GPRMC,111230.00,A,3542.10836,N,05121.17317,E,2.541,22.52,261222,,,A*54 $GPRMC,111236.00,A,3542.11435,N,05121.17364,E,2.751,9.40,261222,,,A*61
35.701806 51.352894 35.70190583333333 51.352894 0.0 0.0
Latitude=35.70190583333333and Longitude=51.352894
Azimuth=2.9979375916586526,Sun Angle=2.9979375916586526
$GPRMC,111237.00,A,3542.11532,N,05121.17365,E,2.398,8.56,261222,,,A*60 $GPRMC,111242.00,A,3542.12017,N,05121.17084,E,2.505,317.13,261222,,,A*61
35.701922 51.35284733333334 35.70200283333333 51.35284733333334 0.0 0.0
Latitude=35.70200283333333and Longitude=51.35284733333334
Azimuth=2.9960879922471624,Sun Angle=2.9960879922471624
$GPRMC,111243.00,A,3542.12058,N,05121.16994,E,2.593,310.12,261222,,,A*68 $GPRMC,111249.00,A,3542.12220,N,05121.16553,E,2.267,,261222,,,A*78
35.70200966666667 51.35275883333333 35.70203666666666 51.35275883333333 0.0 0.0
Latitude=35.70203666666666and Longitude=51.35275883333333
Azimuth=2.993938765432296,Sun Angle=2.993938765432296
$GPRMC,111250.00,A,3542.12216,N,05121.16470,E,2.725,,261222,,,A*76 $GPRMC,111255.00,A,3542.12278,N,05121.16098,E,2.386,271.32,261222,,,A*6F
35.702036 51.352683 35.70204633333335 51.352683 0.0 0.0
Latitude=2.9920934211147117,Sun Angle=2.9920934211147117
Azimuth=2.9920934211147117,Sun Angle=2.9920934211147117
$GPRMC,111256.00,A,3542.12281,N,05121.16027,E,2.274,272.98,261222,,,A*60 $GPRMC,111302.00,A,3542.12365,N,05121.15535,E,2.712,272.93,261222,,,A*61
35.70204633333334 51.35258916666667 35.70206083333334 51.35258916666667 0.0 0.0
Latitude=35.70206083333334and Longitude=51.35258916666667
Azimuth=2.9890456649140720,Sun Angle=2.9890456649140720
$GPRMC,111303.00,A,3542.12377,N,05121.15455,E,2.280,277.81,261222,,,A*6C $GPRMC,111309.00,A,3542.12373,N,05121.14949,E,2.354,,261222,,,A*7E
35.70206283333336 51.3524915 35.70206216666664 51.3524915 -0.0 -0.0
Latitude=35.70206216666664and Longitude=51.3524915
Azimuth=2.987799180812416,Sun Angle=2.987799180812416
```

تصویر ۱۵ خروجی سنسور GPS در دانشگاه

```
Azimuth=-2.963629385627599,Sun Angle=2.963629385627599
$GPRMC,111429.00,A,3542.13705,N,05121.09754,E,0.136,,261222,,,A*75 $GPRMC,111434.00,A,3542.13744,N,05121.09730,E,0.641,,261222,,,A*79
35.70228416666666 51.35162166666667 35.70229066666667 51.35162166666667 0.0 0.0
Latitude=35.70229066666667and Longitude=51.35162166666667
Azimuth=-2.961795034894503,Sun Angle=2.961795034894503
$GPRMC,111435.00,A,3542.13746,N,05121.09739,E,0.044,,261222,,,A*70 $GPRMC,111441.00,A,3542.13698,N,05121.09702,E,0.174,,261222,,,A*7B
35.702291 51.351617 35.702283 51.351617 -0.0 -0.0
Latitude=35.702283and Longitude=51.351617
Azimuth=-2.959656317221534,Sun Angle=2.959656317221534
$GPRMC,111442.00,A,3542.13691,N,05121.09694,E,0.109,,261222,,,A*75 $GPRMC,111448.00,A,3542.13670,N,05121.09548,E,0.341,,261222,,,A*7C
35.70228183333333 51.35159133333333 35.70227833333333 51.35159133333333 -0.0 -0.0
Latitude=35.70227833333333and Longitude=51.35159133333333
Azimuth=-2.9572097427525,Sun Angle=2.9572097427525
$GPRMC,111449.00,A,3542.13659,N,05121.09540,E,0.054,,261222,,,A*7F $GPRMC,111454.00,A,3542.13841,N,05121.09466,E,2.552,341.91,261222,,,A*66
35.70227649090909 51.35157766666664 35.70230683333333 51.35157766666664 0.0 0.0
Latitude=35.70230683333333and Longitude=51.35157766666664
Azimuth=-2.955689490899079,Sun Angle=2.955689490899079
$GPRMC,111455.00,A,3542.13901,N,05121.09463,E,2.659,339.99,261222,,,A*68 $GPRMC,111501.00,A,3542.14318,N,05121.09407,E,2.438,,261222,,,A*7D
35.70231683333333 51.35156783333334 35.70238633333334 51.35156783333334 0.0 0.0
Latitude=35.70238633333334and Longitude=51.35156783333334
Azimuth=-2.953558962019554,Sun Angle=2.953558962019554
$GPRMC,111502.00,A,3542.14378,N,05121.09373,E,2.656,348.45,261222,,,A*66 $GPRMC,111508.00,A,3542.14645,N,05121.09253,E,2.214,,261222,,,A*76
35.70239633333333 51.35154216666667 35.70244083333334 51.35154216666667 0.0 0.0
Latitude=35.70244083333334and Longitude=51.35154216666667
Azimuth=-2.9514295924119547,Sun Angle=2.9514295924119547
$GPRMC,111509.00,A,3542.14677,N,05121.09177,E,1.881,347.66,261222,,,A*68 $GPRMC,111515.00,A,3542.14740,N,05121.08998,E,2.658,302.13,261222,,,A*62
35.70244616666667 51.35149966666667 35.70245666666667 51.35149966666667 0.0 0.0
Latitude=35.70245666666667and Longitude=51.35149966666667
Azimuth=-2.949299218024463,Sun Angle=2.949299218024463
$GPRMC,111516.00,A,3542.14746,N,05121.08942,E,2.301,299.06,261222,,,A*6E $GPRMC,111522.00,A,3542.14355,N,05121.08777,E,4.751,213.14,261222,,,A*61
35.70245766666667 51.35146283333336 35.7023925 51.35146283333336 -0.0 -0.0
Latitude=35.7023925and Longitude=51.35146283333336
Azimuth=-2.9471709014548595,Sun Angle=2.9471709014548595
$GPRMC,111523.00,A,3542.14221,N,05121.08870,E,5.405,,261222,,,A*72 $GPRMC,111529.00,A,3542.14016,N,05121.09280,E,3.323,,261222,,,A*7F
35.70237016666667 51.35154666666664 35.702336 51.35154666666664 -0.0 -0.0
Latitude=35.702336and Longitude=51.35154666666664
Azimuth=-2.9450373886078425,Sun Angle=2.9450373886078425
$GPRMC,111530.00,A,3542.14066,N,05121.09439,E,4.207,,261222,,,A*74 $GPRMC,111536.00,V,,,,,,261222,,,N*7B
35.70234433333336 0.0 0.0 0.0 -0.0 -0.0
Latitude=0.0and Longitude=0.0
Azimuth=1.0474488844983487,Sun Angle=1.0474488844983487
^CTraceback (most recent call last):
  File "/home/mmd/Desktop/project/hw_project-group-3/Code/gps.py", line 30, in <module>
    newdata = ser.readline()
  File "/usr/lib/python3/dist-packages/serial/serialposix.py", line 547, in read
    ready, _ = select.select([self.fd, self.pipe_abort_read_r], [], [], timeout.time_left())
KeyboardInterrupt
```

تصویر ۱۶ خروجی سنسور GPS در دانشگاه ۲

با مقایسه این خروجی با مقدار دقیق مختصات آن نقطه، متوجه شدیم خروجی GPS کاملاً صحیح است و فقط با مقداری تاخیر تغییر می‌کند.

## ۸. تشخیص جهت تابش آفتاب به کمک مختصات

در نهایت، به کمک خروجی گرفته شده در بخش قبل می‌خواهیم زاویه تابش آفتاب را بیابیم. بدین منظور، علاوه بر اطلاعات مذکور به تاریخ و زمان نیز نیاز داریم. سپس به کمک تابع `get_position` کتابخانه `suncalc` در پایتون می‌توانیم زاویه تابش آفتاب را بیابیم. این تابع با گرفتن ۳ ورودی، به ترتیب تاریخ (در قالب `datetime` پایتون، که شامل تاریخ و زمان است)، `longitude` (به شکل درجه در قالب `float` پایتون) و `latitude` (به شکل درجه در قالب `float` پایتون) خروجی را در قالب یک دیکشنری می‌دهد، که به ترتیب شامل دو `key` می‌باشد: `azimuth` و `altitude`. محاسبات این کتابخانه بطور خلاصه به کمک ورودی‌های گفته شده، حرکت زمین به دور خورشید که با سرعت ثابتی نیست و زاویه بین محور چرخش زمین و صفحه مدار آن، صورت می‌گیرد.

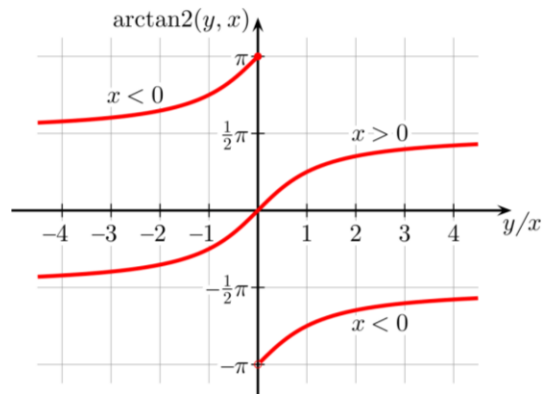
قطعه کد لازم برای این بخش در فایل `gps.py` قابل مشاهده است. برای پیاده‌سازی این بخش از [این لینک](#) کمک گرفتیم.

## ۹. محاسبه جهت حرکت جسم (خودرو)

برای تشخیص محلی از خودرو که در برابر خورشید است نیاز به دانستن جهت حرکت آن داریم. اگر فرض کنیم یک خودروی در حال حرکت در یک لحظه در مختصات  $\begin{bmatrix} lng1 \\ lat1 \end{bmatrix}$  و در لحظه بعد پس از حرکت بدون تغییر جهت در مختصات  $\begin{bmatrix} lng2 \\ lat2 \end{bmatrix}$  قرار دارد، مسافتی که در جهت شمال-جنوب طی کرده است از فرمول  $2R \sin(\frac{lat2-lat1}{2})$  و مسافتی که در جهت شرق-غرب طی کرده است از فرمول  $2R \sin(\frac{lng2-lng1}{2})$  بدست می‌آیند و  $R$  شعاع کره زمین است. در نتیجه، جهت حرکت خودرو نسبت به محور شمال-جنوب از فرمول زیر بدست خواهد آمد:

$$\tan^{-1} \frac{\sin(\frac{lng2 - lng1}{2})}{\sin(\frac{lat2 - lat1}{2})}$$

در نهایت و به کمک تشخیص جهت قرارگیری خودرو و محل خورشید در آسمان، می‌توان محلی از خودرو که در برابر نور خورشید قرار دارد را شناسایی کرد. لازم به ذکر است که برای محاسبه معکوس تانژانت ابتدا از تابع `arctan` در پایتون استفاده می‌کردیم که به دلیل محدود بودن خروجی آن بین منفی ۹۰ تا مثبت ۹۰ درجه، این تابع را به `arctan2` تغییر دادیم که خروجی آن بین منفی ۱۸۰ تا مثبت ۱۸۰ درجه است.



تصویر ۱۷ نمودار تابع atan2

نهایتاً به کمک قطعه کد زیر تشخیص می‌دهیم کدام بخش خودرو در معرض نور خورشید است. بخش‌های ۰ تا ۳ خودرو مطابق بخش‌ها گفته شده در بخش [گرفتن دما](#) هستند.

```
if sun_angle < np.pi / 2:
    exposed_sensor = 2
elif sun_angle < np.pi:
    exposed_sensor = 3
elif sun_angle < 3 * np.pi / 2:
    exposed_sensor = 1
else:
    exposed_sensor = 0
```

تصویر ۱۸ تشخیص بخش در معرض آفتاب

## ۱۰. صحت‌سنجی سنسور GPS

برای بررسی کارکرد این سنسور به شرایط خاصی نیاز داشتیم. به دلیل ضعیف بودن سیگنال‌های GPS نمی‌توانستیم در محیط بسته آن را تست کنیم و ناچار بودیم در محیط آزاد مانند حیاط دانشگاه به تست آن بپردازیم. هم‌چنین برای بررسی تغییر مختصات GPS باید حرکت می‌کردیم، اما سرعت حرکت با پیاده‌روی بسیار پایین بود و تغییر چشمگیری در مختصات دیده نمی‌شد. به همین دلایل، پس از اطمینان از خروجی صحیح این سنسور، برای تسهیل تست سایر بخش‌های پروژه از یک تابع به نام `test generate` کردن مختصات استفاده کردیم. در این تابع مختصات یک نقطه ابتدایی (مثلاً نقطه خاصی از دانشگاه) در قالب `longitude` و `latitude`، جهتی که می‌خواهیم حرکت را در آن تست کنیم به شکل `N`، `E`، `S` و `W`، و یک مقدار ثابت گام برای تغییر مختصات، بطور پیش‌فرض `0.001`، به عنوان ورودی داده می‌شود و با اضافه یا کم کردن گام از طول و عرض جغرافیایی بر حسب جهت گفته شده، مختصات‌های جدید `generate` می‌گردند. با این کار می‌توانیم برای صحت‌سنجی جهت تابش آفتاب در خودرو و روشن کردن LED مربوطه، به راحتی داده تولید کنیم. به عنوان مثال، برای بررسی کد بخش [جهت تابش آفتاب بر اساس جهت حرکت خودرو](#) از این داده‌سازی استفاده کردیم و خروجی آن در تصاویر زیر قابل مشاهده است:

```
mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ nano gps.py
mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ python gps.py
35.703129 51.351671 35.704128999999995 51.351671 0.0 0.000999999833331011
Movement Directio: 0.0 ,Sun Azimuth: 1.22794032588571 ,Sun Angle: 1.22794032588571
Sensor 2 is close to the Sun.
35.704128999999995 51.351671 35.705128999999999 51.351671 0.0 0.000999999833331011
Movement Directio: 0.0 ,Sun Azimuth: 1.2281352125737597 ,Sun Angle: 1.2281352125737597
Sensor 2 is close to the Sun.
^CTraceback (most recent call last):
  File "/home/mmd/Desktop/project/hw_project-group-3/Code/gps.py", line 107, in <module>
    manual_main()
  File "/home/mmd/Desktop/project/hw_project-group-3/Code/gps.py", line 102, in manual_main
    time.sleep(5)
KeyboardInterrupt
```

تصویر ۱۹ تشخیص محل تابش آفتاب در صورت حرکت به سمت شمال

```
mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ nano gps.py
mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ python gps.py
35.703129 51.351671 35.703129 51.352671 0.000999999833331011 0.0
Movement Directio: 1.5707963267948966 ,Sun Azimuth: 1.231415676012904 ,Sun Angle: 5.943804656397594
Sensor 0 is close to the Sun.
35.703129 51.352671 35.703129 51.353671 0.000999999833331011 0.0
Movement Directio: 1.5707963267948966 ,Sun Azimuth: 1.2316159058045542 ,Sun Angle: 5.944004886189244
Sensor 0 is close to the Sun.
^CTraceback (most recent call last):
  File "/home/mmd/Desktop/project/hw_project-group-3/Code/gps.py", line 107, in <module>
    manual_main()
  File "/home/mmd/Desktop/project/hw_project-group-3/Code/gps.py", line 102, in manual_main
    time.sleep(5)
KeyboardInterrupt
```

تصویر ۲۰ تشخیص محل تابش آفتاب در صورت حرکت به سمت شرق

```
mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ nano gps.py
mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ python gps.py
35.703129 51.351671 35.702129 51.351671 0.0 -0.000999999833331011
Movement Directio: 3.141592653589793 ,Sun Azimuth: 1.2324272859094225 ,Sun Angle: 4.374019939499216
Sensor 1 is close to the Sun.
35.702129 51.351671 35.701129 51.351671 0.0 -0.000999999833331011
Movement Directio: 3.141592653589793 ,Sun Azimuth: 1.2326146470065626 ,Sun Angle: 4.374207300596356
Sensor 1 is close to the Sun.
^CTraceback (most recent call last):
  File "/home/mmd/Desktop/project/hw_project-group-3/Code/gps.py", line 107, in <module>
    manual_main()
  File "/home/mmd/Desktop/project/hw_project-group-3/Code/gps.py", line 102, in manual_main
    time.sleep(5)
KeyboardInterrupt
```

تصویر ۲۱ تشخیص محل تابش آفتاب در صورت حرکت به سمت جنوب

```
mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ nano gps.py
mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ python gps.py
35.703129 51.351671 35.703129 51.350671000000006 -0.000999999833331011 0.0
Movement Directio: -1.5707963267948966 ,Sun Azimuth: 1.233371821130741 ,Sun Angle: 2.8041681479256377
Sensor 3 is close to the Sun.
35.703129 51.350671000000006 35.703129 51.349671000000001 -0.000999999833331011 0.0
Movement Directio: -1.5707963267948966 ,Sun Azimuth: 1.2335535684602756 ,Sun Angle: 2.8043498952551724
Sensor 3 is close to the Sun.
^CTraceback (most recent call last):
  File "/home/mmd/Desktop/project/hw_project-group-3/Code/gps.py", line 107, in <module>
    manual_main()
  File "/home/mmd/Desktop/project/hw_project-group-3/Code/gps.py", line 102, in manual_main
    time.sleep(5)
KeyboardInterrupt
```

تصویر ۲۲ تشخیص محل تابش آفتاب در صورت حرکت به سمت غرب

## ۱.۱ روشن و خاموش کردن خنک‌کننده‌ها

همانطور که گفته شد، به جای خنک‌کننده از LED استفاده می‌کنیم. برای ۴ بخش خودرو به ۴ LED نیاز داریم و هر یک باید با یک سنسور دما هماهنگ باشد، بطوریکه پس از تشخیص جهت تابش آفتاب، با اندازه‌گیری دمای ۴ بخش خودرو، در صورت گرم بودن هر بخش چراغ LED آن روشن شود. برای این کار ۴ LED را به ۴ پین GPIO مجزا از رزبری پای متصل می‌کنیم. با این کار



به هر دماسنج یک خنک‌کننده تخصیص می‌دهیم. سپس در هر ثانیه دمای یک سنسور را دریافت می‌کنیم و در نتیجه، پس از ۴ ثانیه ۴ دمای جدید سنسورها گرفته می‌شود. اکنون میانگین آنها را محاسبه کرده و بخش‌هایی از خودرو که دمایی ۱ درجه بیش از میانگین دارند را به عنوان بخش‌هایی که باید خنک شوند در نظر می‌گیریم و چراغ بخش مربوطه را روشن می‌کنیم. LEDهای بخش‌های ۰، ۱، ۲ و ۳ خودرو به ترتیب به GPIOهای ۵، ۶، ۱۳ و ۱۹ رزبری پای متصلند. در پیاده‌سازی این بخش به چند تابع ساده محاسبه میانگین و ... نیاز داریم که در فایل `math_functions.py` قرار دارد. کارکرد این بخش از کد تست شده و در تصاویر زیر قابل مشاهده است:

```
mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ python temp_sensor.py
Sensor 1 => 23.625 (C) - 74.525 (F)
Sensor 2 => 23.625 (C) - 74.525 (F)
Sensor 3 => 23.812 (C) - 74.86160000000001 (F)
Sensor 4 => 23.562 (C) - 74.4116 (F)
LED 0 off
LED 1 off
LED 2 off
LED 3 off
Sensor 1 => 23.562 (C) - 74.4116 (F)
Sensor 2 => 23.625 (C) - 74.525 (F)
Sensor 3 => 23.75 (C) - 74.75 (F)
Sensor 4 => 23.562 (C) - 74.4116 (F)
LED 0 off
LED 1 off
LED 2 off
LED 3 off
Sensor 1 => 28.062 (C) - 82.5116 (F)
Sensor 2 => 23.625 (C) - 74.525 (F)
Sensor 3 => 23.812 (C) - 74.86160000000001 (F)
Sensor 4 => 23.625 (C) - 74.525 (F)
LED 0 on
LED 1 off
LED 2 off
LED 3 off
Sensor 1 => 29.25 (C) - 84.65 (F)
Sensor 2 => 23.75 (C) - 74.75 (F)
Sensor 3 => 23.937 (C) - 75.0866 (F)
Sensor 4 => 23.687 (C) - 74.6366 (F)
LED 0 on
LED 1 off
LED 2 off
```

تصویر ۲۳ صحت‌سنجی تطابق سنسور دما و LED

```
LED 2 off
LED 3 off
Sensor 1 => 29.625 (C) - 85.325 (F)
Sensor 2 => 23.75 (C) - 74.75 (F)
Sensor 3 => 24.0 (C) - 75.2 (F)
Sensor 4 => 23.687 (C) - 74.6366 (F)
LED 0 on
LED 1 off
LED 2 off
LED 3 off
Sensor 1 => 29.812 (C) - 85.66159999999999 (F)
Sensor 2 => 23.75 (C) - 74.75 (F)
Sensor 3 => 24.0 (C) - 75.2 (F)
Sensor 4 => 23.687 (C) - 74.6366 (F)
LED 0 on
LED 1 off
LED 2 off
LED 3 off
Sensor 1 => 29.937 (C) - 85.8866 (F)
Sensor 2 => 23.75 (C) - 74.75 (F)
Sensor 3 => 24.0 (C) - 75.2 (F)
Sensor 4 => 23.75 (C) - 74.75 (F)
LED 0 on
LED 1 off
LED 2 off
LED 3 off
Sensor 1 => 30.0 (C) - 86.0 (F)
Sensor 2 => 28.312 (C) - 82.9616 (F)
Sensor 3 => 24.0 (C) - 75.2 (F)
Sensor 4 => 23.75 (C) - 74.75 (F)
LED 0 on
LED 1 on
LED 2 off
LED 3 off
Sensor 1 => 30.0 (C) - 86.0 (F)
Sensor 2 => 28.812 (C) - 83.8616 (F)
Sensor 3 => 24.862 (C) - 75.3116 (F)
Sensor 4 => 23.75 (C) - 74.75 (F)
LED 0 on
LED 1 on
LED 2 off
LED 3 off
Sensor 1 => 28.625 (C) - 83.525 (F)
Sensor 2 => 27.0 (C) - 80.6 (F)
Sensor 3 => 24.062 (C) - 75.3116 (F)
Sensor 4 => 23.75 (C) - 74.75 (F)
LED 0 on
LED 1 on
LED 2 off
```

تصویر ۲۴ صحت‌سنجی تطابق سنسور دما و LED ۲

```

LED 2 off
LED 3 off
Sensor 1 => 27.125 (C) - 80.825 (F)
Sensor 2 => 25.937 (C) - 78.6866 (F)
Sensor 3 => 24.062 (C) - 75.3116 (F)
Sensor 4 => 23.75 (C) - 74.75 (F)
LED 0 on
LED 1 off
LED 2 off
LED 3 off
Sensor 1 => 26.187 (C) - 79.1366 (F)
Sensor 2 => 25.25 (C) - 77.45 (F)
Sensor 3 => 24.0 (C) - 75.2 (F)
Sensor 4 => 23.687 (C) - 74.6366 (F)
LED 0 on
LED 1 off
LED 2 off
LED 3 off
Sensor 1 => 25.625 (C) - 78.125 (F)
Sensor 2 => 24.687 (C) - 76.4366 (F)
Sensor 3 => 23.937 (C) - 75.0866 (F)
Sensor 4 => 23.625 (C) - 74.525 (F)
LED 0 on
LED 1 off
LED 2 off
LED 3 off
Sensor 1 => 25.187 (C) - 77.3366 (F)
Sensor 2 => 24.5 (C) - 76.1 (F)
Sensor 3 => 23.937 (C) - 75.0866 (F)
Sensor 4 => 23.687 (C) - 74.6366 (F)
LED 0 off
LED 1 off
LED 2 off
LED 3 off
Sensor 1 => 24.937 (C) - 76.8866 (F)
Sensor 2 => 24.375 (C) - 75.875 (F)
Sensor 3 => 23.937 (C) - 75.0866 (F)
Sensor 4 => 23.687 (C) - 74.6366 (F)
LED 0 off
LED 1 off
LED 2 off
LED 3 off
Sensor 1 => 24.687 (C) - 76.4366 (F)
Sensor 2 => 24.25 (C) - 75.65 (F)
Sensor 3 => 23.937 (C) - 75.0866 (F)
Sensor 4 => 23.625 (C) - 74.525 (F)
LED 0 off
LED 1 off
LED 2 off

```

تصویر ۲۵ صحت‌سنجی تطابق سنسور دما و LED ۳

## ۱۲. بخش اصلی پروژه

نهایتاً در قطعه کد زیر در فایل temp\_sensor.py به اجرای پروژه می‌پردازیم. طبق این کد، در بخش‌هایی از خودرو که دمایی بیش از یک درجه بالای میانگین دمای همه بخش‌های خودرو دارند، در صورت در معرض نور آفتاب بودن، خنک‌کننده روشن می‌شود؛ و در غیر این‌صورت خاموش می‌گردد.

```

for sensor_id in hot_sensors_ids:
    if sensor_id == exposed_sensor:
        turn_on_led(led_temp_sensor[sensor_id])
    else:
        turn_off_led(led_temp_sensor[sensor_id])

```

تصویر ۲۶ کد ترکیب شرط دما و تابش آفتاب

```

pi@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ python gps.py
35.703129 51.351671 35.703129 51.352671 0.00099999983331011 0.0
Movement Direction: 1.5707963267948966 ,Sun Azimuth: -0.8911995183938662 ,Sun Angle: 3.8191894619916233
Sensor 1 is close to the Sun.
Sensor 0 => 23.375 (C) - 74.075 (F)
Sensor 1 => 23.625 (C) - 74.525 (F)
Sensor 2 => 26.0 (C) - 78.8 (F)
Sensor 3 => 23.937 (C) - 75.0866 (F)
LED 2 off
LED 0 off
LED 1 off
LED 3 off
35.703129 51.352671 35.703129 51.353671 0.00099999983331011 0.0
Movement Direction: 1.5707963267948966 ,Sun Azimuth: -0.8928022786913962 ,Sun Angle: 3.8195867016932934
Sensor 1 is close to the Sun.
Sensor 0 => 23.375 (C) - 74.075 (F)
Sensor 1 => 24.312 (C) - 75.7616 (F)
Sensor 2 => 25.625 (C) - 78.125 (F)
Sensor 3 => 23.937 (C) - 75.0866 (F)
LED 2 off
LED 0 off
LED 1 off
LED 3 off
35.703129 51.353671 35.703129 51.354670999999996 0.00099999983331011 0.0
Movement Direction: 1.5707963267948966 ,Sun Azimuth: -0.8924007498123694 ,Sun Angle: 3.8199882305723203
Sensor 1 is close to the Sun.
Sensor 0 => 23.375 (C) - 74.075 (F)
Sensor 1 => 27.25 (C) - 81.95 (F)
Sensor 2 => 25.375 (C) - 77.675 (F)
Sensor 3 => 23.937 (C) - 75.0866 (F)
LED 1 on
LED 0 off
LED 2 off
LED 3 off
35.703129 51.354670999999996 35.703129 51.355670999999994 0.00099999983331011 0.0
Movement Direction: 1.5707963267948966 ,Sun Azimuth: -0.8920070276487032 ,Sun Angle: 3.8203819527359864
Sensor 1 is close to the Sun.
Sensor 0 => 23.437 (C) - 74.1866 (F)

```

تصویر ۲۷ نتیجه نهایی اجرای کد

```

pi@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ python gps.py
35.703129 51.351671 35.703129 51.352671 0.00099999983331011 0.0
Movement Direction: 1.5707963267948966 ,Sun Azimuth: -0.8888704147978949 ,Sun Angle: 3.8235185655867947
Sensor 1 is close to the Sun.
Sensor 0 => 23.562 (C) - 74.4116 (F)
Sensor 1 => 24.562 (C) - 76.2116 (F)
Sensor 2 => 24.125 (C) - 75.425 (F)
Sensor 3 => 24.062 (C) - 75.3116 (F)
LED 0 off
LED 1 off
LED 2 off
LED 3 off
35.703129 51.352671 35.703129 51.353671 0.00099999983331011 0.0
Movement Direction: 1.5707963267948966 ,Sun Azimuth: -0.8884731403490301 ,Sun Angle: 3.8239158400356597
Sensor 1 is close to the Sun.
Sensor 0 => 23.562 (C) - 74.4116 (F)
Sensor 1 => 24.5 (C) - 76.1 (F)
Sensor 2 => 24.062 (C) - 75.3116 (F)
Sensor 3 => 26.562 (C) - 79.8116 (F)
LED 3 off
LED 0 off
LED 1 off
LED 2 off
35.703129 51.353671 35.703129 51.354670999999996 0.00099999983331011 0.0
Movement Direction: 1.5707963267948966 ,Sun Azimuth: -0.8880741964503484 ,Sun Angle: 3.824314783934341
Sensor 1 is close to the Sun.
Sensor 0 => 23.625 (C) - 74.525 (F)
Sensor 1 => 28.375 (C) - 83.075 (F)
Sensor 2 => 24.0 (C) - 75.2 (F)
Sensor 3 => 28.562 (C) - 83.41159999999999 (F)
LED 1 on
LED 3 off
LED 0 off
LED 2 off
35.703129 51.354670999999996 35.703129 51.355670999999994 0.00099999983331011 0.0
Movement Direction: 1.5707963267948966 ,Sun Azimuth: -0.8876709325765141 ,Sun Angle: 3.8247180478081755
Sensor 1 is close to the Sun.
Sensor 0 => 23.625 (C) - 74.525 (F)
Sensor 1 => 29.862 (C) - 84.3116 (F)
Sensor 2 => 24.0 (C) - 75.2 (F)
Sensor 3 => 29.862 (C) - 84.3116 (F)
LED 1 on
LED 3 off
LED 0 off
LED 2 off
35.703129 51.355670999999994 35.703129 51.35667099999999 0.00099999983331011 0.0
Movement Direction: 1.5707963267948966 ,Sun Azimuth: -0.8872755422863866 ,Sun Angle: 3.825113438098303
Sensor 1 is close to the Sun.

```

تصویر ۲۸ نتیجه نهایی اجرای کد ۲

```

Movement Direction: 1.5707963267948966 ,Sun Azimuth: -0.8775706106309099 ,Sun Angle: 3.83481836975378
Sensor 1 is close to the Sun.
Sensor 0 => 23.625 (C) - 74.525 (F)
Sensor 1 => 26.5 (C) - 79.7 (F)
Sensor 2 => 23.5 (C) - 74.1 (F)
Sensor 3 => 24.375 (C) - 75.875 (F)
LED 1 on
LED 0 off
LED 2 off
LED 3 off
35.703129 51.356670999999994 35.703129 51.356670999999999 0.00099999983331011 0.0
Movement Direction: 1.5707963267948966 ,Sun Azimuth: -0.8771635478395572 ,Sun Angle: 3.8352254325451325
Sensor 1 is close to the Sun.
Sensor 0 => 23.625 (C) - 74.525 (F)
Sensor 1 => 25.75 (C) - 78.35 (F)
Sensor 2 => 23.562 (C) - 74.4116 (F)
Sensor 3 => 24.375 (C) - 75.875 (F)
LED 1 on
LED 0 off
LED 2 off
LED 3 off
35.703129 51.356670999999999 35.703129 51.357670999999999 0.00099999983331011 0.0
Movement Direction: 1.5707963267948966 ,Sun Azimuth: -0.8767644300983511 ,Sun Angle: 3.8356245502863380
Sensor 1 is close to the Sun.
Sensor 0 => 23.625 (C) - 74.525 (F)
Sensor 1 => 25.25 (C) - 77.45 (F)
Sensor 2 => 23.562 (C) - 74.4116 (F)
Sensor 3 => 24.375 (C) - 75.875 (F)
LED 1 on
LED 0 off
LED 2 off
LED 3 off
35.703129 51.357670999999999 35.703129 51.358670999999999 0.00099999983331011 0.0
Movement Direction: 1.5707963267948966 ,Sun Azimuth: -0.8763651513804178 ,Sun Angle: 3.836023829004272
Sensor 1 is close to the Sun.
Sensor 0 => 23.562 (C) - 74.4116 (F)
Sensor 1 => 25.0 (C) - 77.0 (F)
Sensor 2 => 23.625 (C) - 74.525 (F)
Sensor 3 => 24.375 (C) - 75.875 (F)
LED 0 off
LED 1 off
LED 2 off
LED 3 off
35.703129 51.358670999999999 35.703129 51.359670999999999 0.00099999983331011 0.0
Movement Direction: 1.5707963267948966 ,Sun Azimuth: -0.8759576487935229 ,Sun Angle: 3.8364313315911667
Sensor 1 is close to the Sun.
*GTraceback (most recent call last):
  File "/home/rmd/Desktop/project/bu_project-group-3/Code/gps.py", line 120, in <module>

```

تصویر ۲۹ نتیجه نهایی اجرای کد ۳