



گزارش میانی سوم آزمایشگاه سخت افزار

گروه سه

اعضا:

بهار خدابخشیان ۹۷۱۰۵۹۰۶

محمدامین شریفی ۹۷۱۱۰۰۵۸

درنا دهقانی ۹۷۱۰۵۹۳۹

شرح فعالیت‌ها

فعالیت‌های این بخش از پروژه شامل دریافت اطلاعات لازم از سنسور GPS و استفاده از آنها بر مبنای نیاز پروژه بود. سنسور Neo-6m-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:

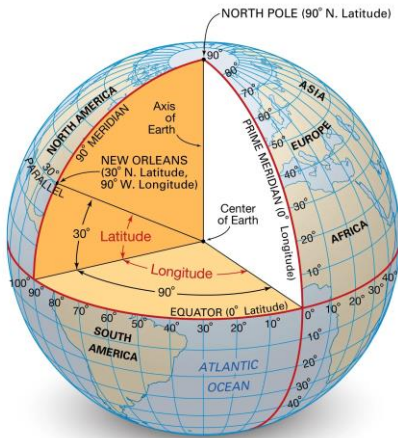
```
mmd@raspberrypi:~/Desktop/project/hw_project-group-3/Code $ python gps.py
$GPRMC,105506.00,V,,,,,,181222,,N*70
{'azimuth': -0.5518667376714196, 'altitude': 1.0859745784966284}
Latitude=0.0and Longitude=0.0
$GPRMC,105507.00,V,,,,,,181222,,N*71
{'azimuth': -0.5517680835033895, 'altitude': 1.0860065621303725}
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 ۱ خروجی سنسور و کد GPS

این اطلاعات به ترتیب عبارتند از:

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

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:

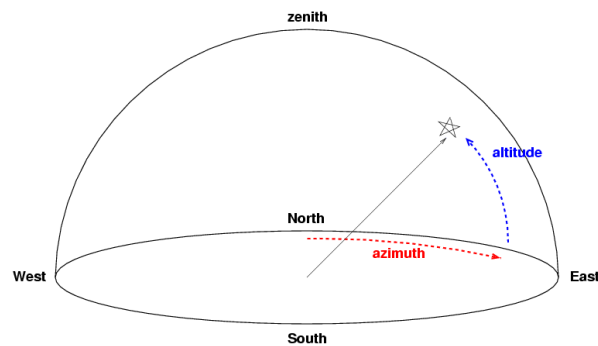


© Encyclopædia Britannica, Inc.

تصویر ۲ نمایش عرض و طول جغرافیایی

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.



تصویر ۳ نمایش دستگاه مختصات افقی

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 برحسب درجه از افق که از ۰ تا ۹۰ درجه تغییر می‌کند.

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

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 $[lng1]$ و در لحظه بعد پس از حرکت بدون تغییر جهت در مختصات $[lng2]$ 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

بدست می‌آیند و R شعاع کره زمین است. در نتیجه، جهت حرکت خودرو نسبت به محور شمال-جنوب از فرمول زیر بدست خواهد آمد:

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

```

raspberrypi:~/Desktop/project/hw_project-group-3/code % cat gps.py
import serial
import time
import string
import pymea2
import numpy as np
from sunalc import get_position
from datetime import datetime
# import astropy.coordinates as coord
# from astropy.time import Time
# import astropy.units as u

def calculate_solar_angle(latitude, longitude):
    loc = coord.EarthLocation(lon=longitude * u.deg, lat=latitude * u.deg)
    now = Time.now()
    altaz = coord.AltAz(location=loc, obstime=now)
    sun = coord.get_sun(now)
    return sun.transform_to(altaz).alt

while True:
    port = "/dev/ttyAMA0"
    ser = serial.Serial(port, baudrate=9600, timeout=0.5)
    dataout = pymea2.MPEASTreamReader()
    data1 = ser.readline()
    current_time = time.time()

    if data1[0:6] == b'$GPRMC':
        data2 = None
        while time.time() - current_time < 5:
            newdata = ser.readline()
            if newdata[0:6] == b'$GPRMC' and time.time() - current_time > 1:
                data2 = newdata
                break

```

تصویر ۴ کد تشخیص جهت تابش خورشید به خودرو ۱

```

if not data2:
    print("timeout!")
    continue

data1 = str(data1)[2:-5]
data2 = str(data2)[2:-5]

msg1 = pymea2.parse(data1)
msg2 = pymea2.parse(data2)

lat1 = msg1.latitude
lng1 = msg1.longitude

lat2 = msg2.latitude
lng2 = msg2.longitude

angle_from_north = np.arctan(np.sin((lng2 - lng1) / 2) / np.sin((lat2 - lat1) / 2))

date = datetime.now()
sun_pos = get_position(date, lng2, lat2)

loc = "Latitude=" + str(lat2) + " and Longitude=" + str(lng2) + "\nAzimuth=" + str(sun_pos[azimuth]) + "Sun Angle=" + str(angle_from_north * 180 / np.pi - sun_pos[azimuth])
print(loc)

```

تصویر ۵ کد تشخیص جهت تابش خودرو ۲

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.

کارکرد سنسور GPS

همانطور که 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 ۱ ثانیه مجدداً مختصات جدیدی دریافت می‌گردد تا به کمک این دو مقدار جهت حرکت خودرو تعیین گردد. مقدار زمان ۱ 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. لازم به ذکر است بنا به نتایج روند تست، این مدت زمان ممکن است افزایش یا کاهش یابد. همچنین اگر به هر علتی، از جمله ضعیف بودن سیگنال GPS، داده‌ی دوم پس از ۱ ثانیه از سنسور گرفته نشد، با هشدار Timeout مواجه می‌شویم و مجدداً تلاش به دریافت این دو مختصات می‌کنیم.