

GPS 模块解析位置信息

1. 学习目标

本次课程我们主要学习使用 jetson nano 和 GPS 模块实读取并解析位置信息。

2. 课前准备

GPS 模块采用的是 UART 通讯或 USB 通讯，这里以 USB 通讯为例。

使用 type-c 线连接 jetson nano 和 GPS 模块，运行命令 `ls /dev | grep 'ttyUSB'`，可以看到识别到 GPS 模块为 USB0

```
jetson@jetson-desktop:~$ ls /dev | grep 'ttyUSB'
ttyUSB0
```

3. 程序

本次课程的程序请参考：GPS.py

初始化 USB:

```
ser = serial.Serial("/dev/ttyUSB0", 9600)
```

位置信息获取和解析函数，下图中在位置信息中筛选出 GNGGA 开头的信息，然后解析出数据存到各个全局变量中。

```

def GPS_read():
    global utctime
    global lat
    global ulat
    global lon
    global ulon
    global numSv
    global msl
    global cogt
    global cogm
    global sog
    global kph
    global gps_t
    if ser.inWaiting():
        if ser.read(1) == b'G':
            if ser.inWaiting():
                if ser.read(1) == b'N':
                    if ser.inWaiting():
                        choice = ser.read(1)
                        if choice == b'G':
                            if ser.inWaiting():
                                if ser.read(1) == b'G':
                                    if ser.inWaiting():
                                        if ser.read(1) == b'A':
                                            #utctime = ser.read(7)
                                            GGA = ser.read(70)
                                            GGA_g = re.findall(r"\w+(?=\,)|(?<=,)\w+", str(GGA))
                                            # print(GGA_g)
                                            if len(GGA_g) < 13:
                                                print("GPS no found")
                                                gps_t = 0
                                                return 0
                                            else:
                                                utctime = GGA_g[0]
                                                # lat = GGA_g[2][0]+GGA_g[2][1]+'°'+GGA_g[2][2]+GGA_g[2][3]+'.'+GGA_g[3]+'°'
                                                lat = "%.8f" % Convert_to_degrees(str(GGA_g[2]), str(GGA_g[3]))
                                                ulat = GGA_g[4]
                                                # lon = GGA_g[5][0]+GGA_g[5][1]+GGA_g[5][2]+'°'+GGA_g[5][3]+GGA_g[5][4]+'.'+GGA_g[6]+'°'
                                                lon = "%.8f" % Convert_to_degrees(str(GGA_g[5]), str(GGA_g[6]))
                                                ulon = GGA_g[7]
                                                numSv = GGA_g[9]
                                                msl = GGA_g[12]+'.'+GGA_g[13]+GGA_g[14]
                                                #print(GGA_g)
                                                gps_t = 1
                                                return 1
    else:
        return 0

def Convert_to_degrees(in_data1, in_data2):
    len_data1 = len(in_data1)
    str_data2 = "%05d" % int(in_data2)
    temp_data = int(in_data1)
    symbol = 1
    if temp_data < 0:
        symbol = -1
    degree = int(temp_data / 100.0)
    str_decimal = str(in_data1[len_data1-2]) + str(in_data1[len_data1-1]) + str(str_data2)
    f_degree = int(str_decimal)/60.0/100000.0
    # print("f_degree:", f_degree)
    if symbol > 0:
        result = degree + f_degree
    else:
        result = degree - f_degree
    return result

```

同样的方法获取了 GNVTG 的航向信息并解析。

```

elif choice == b'V':
    if ser.inWaiting():
        if ser.read(1) == b'T':
            if ser.inWaiting():
                if ser.read(1) == b'G':
                    VTG = ser.read(40)
                    VTG_g = re.findall(r"\w+(?=(,)|(?<=,)\w+", str(VTG))
                    cogt = VTG_g[0]+'.'+VTG_g[1]+'T'
                    if VTG_g[3] == 'M':
                        cogm = '0.00'
                        sog = VTG_g[4]+'.'+VTG_g[5]
                        kph = VTG_g[7]+'.'+VTG_g[8]
                    elif VTG_g[3] != 'M':
                        cogm = VTG_g[3]+'.'+VTG_g[4]
                        sog = VTG_g[6]+'.'+VTG_g[7]
                        kph = VTG_g[9]+'.'+VTG_g[10]
                    #print(kph)

```

解析后的数据循环打印

```

if GPS_read():
    print("*****")
    print('UTC Time:'+utctime)
    print('Latitude:'+lat+ulat)
    print('Longitude:'+lon+ulon)
    print('Number of satellites:'+numSv)
    print('Altitude:'+msl)
    print('True north heading:'+cogt+'°')
    print('Magnetic north heading:'+cogm+'°')
    print('Ground speed:'+sog+'Kn')
    print('Ground speed:'+kph+'Km/h')
    print("*****")

```

4. 运行程序

终端输入 `sudo python3 GPS.py` 运行程序。

5. 实验现象

模块通电后，需要 32s 左右的时间启动，之后模块上的串口打印状态灯会持续闪烁，此时可以正常接收数据。

程序运行以后，开始初始化 USB，初始化成功显示“GPS Serial Opened! Baudrate=9600”，否则显示 “GPS Serial Open Failed!”，如果错误需要

检查接线或 USB 端口，之后循环打印位置和航向信息。

```
jetson@jetson-desktop:~$ sudo python3 GPS.py
GPS Serial Opened! Baudrate=9600
*****
UTC Time:083021
Latitude:22.58311383N
Longitude:113.96576167E
Number of satellites:15
Altitude:31.8M
True north heading:°
Magnetic north heading:°
Ground speed:Kn
Ground speed:Km/h
*****
```

按 Ctrl+C 退出信息读取。

```
*****
UTC Time:083107
Latitude:22.58311783N
Longitude:113.96575433E
Number of satellites:16
Altitude:32.0M
True north heading:94.12T°
Magnetic north heading:0.00°
Ground speed:0.00Kn
Ground speed:0.00Km/h
*****
^CGPS serial Close!
jetson@jetson-desktop:~$
```

注意，模块天线需要在室外，否则可能搜索不到 GPS 信号，搜索不到信号的时候打印"GPS no found"。