StarFive 赛昉科技

Using VisionFive UART to Read GPS Data

with Python

Application Note

Version: 1.1

Date: 2022/07/29

Doc ID: VisionFive-ANEN-008-1.1

Legal Statements

Important legal notice before reading this documentation.

PROPRIETARY NOTICE

Information in this document is provided "as is," with all faults. Contents may be periodically updated or revised due to the product development. Shanghai StarFive Technology Co., Ltd. (hereinafter "StarFive") reserves the right to make changes without further notice to any products herein.

StarFive expressly disclaims all warranties, representations, and conditions of any kind, whether express or implied, including, but not limited to, the implied warranties or conditions of merchantability, fitness for a particular purpose and non-infringement.

StarFive does not assume any liability rising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation indirect, incidental, special, exemplary, or consequential damages.

Contact Us

Address: Room 502, Building 2, No. 61 Shengxia Rd., China (Shanghai) Pilot Free Trade Zone, Shanghai, 201203, China

Website: http://www.starfivetech.com

Email: sales@starfivetech.com(sales) , support@starfivetech.com(support)

Preface

About this guide and technical support information.

About this document

This application note provides steps to use VisionFive's UART to read GPS data through an example program with Python.

Revision History

Table 0-1 Revision History

Version	Released	Revision
V1.1	2022-07-29	Added "cd" in the codeblock `cd /usr/local/lib64/python3.9/site-packages` to make it a complete command.

Notes and notices

The following notes and notices might appear in this guide:

- **Tip:**
  Suggests how to apply the information in a topic or step.
- **Note:**
  Explains a special case or expands on an important point.
- **Important:**
  Points out critical information concerning a topic or step.
- **CAUTION:**
  Indicates that an action or step can cause loss of data, security problems, or performance issues.
- **Warning:**
  Indicates that an action or step can result in physical harm or cause damage to hardware.

Contents

List of Tables

List of Figures

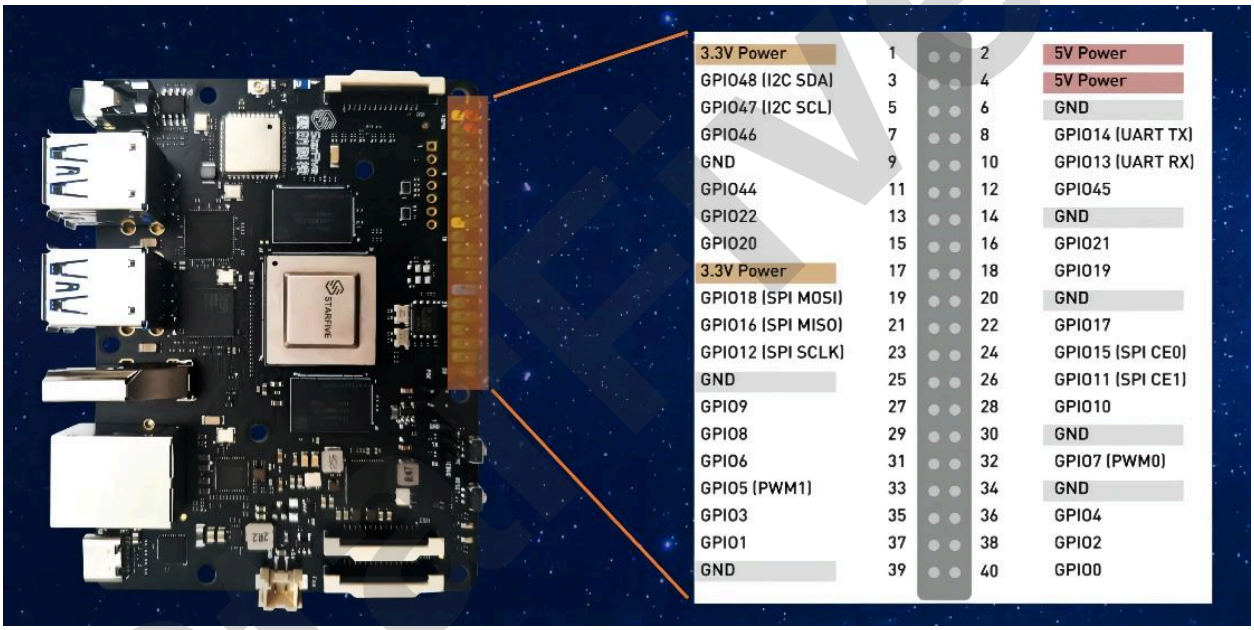
1. Introduction

This application note provides steps to use VisionFive's UART to read GPS data through an example program with Python.

1.1. 40-Pin Header Definition

The following figure shows the location of the 40-pin header. The VisionFive board is taken as an example:

Figure 1-1 40-Pin Definition



2. Preparation

Before executing the demo program, make sure you prepare the following:

2.1. Preparing Hardware

Prepare the following hardware items before running the demo code:

Table 2-1 Hardware Preparation

Type	M/O*	Item	Notes
General	M	StarFive single board computer	The following boards are applicable: • StarLight • VisionFive
General	M	• 16 GB (or more) micro-SD card • micro-SD card reader • Computer (Windows/MAC/Linux) • USB to serial converter (3.3 V I/O) • Ethernet cable • Power adapter (5 V / 3 A) • USB Type-C Cable	These items are used for flashing Fedora OS into a micro-SD card.
UART Demo	M	• NEO-6M GPS • 4 Dupont lines (female to female) • An external antenna (Optional)	The antenna is used to improve GPS signal reception.

Note:
*: M: Mandatory, O: Optional

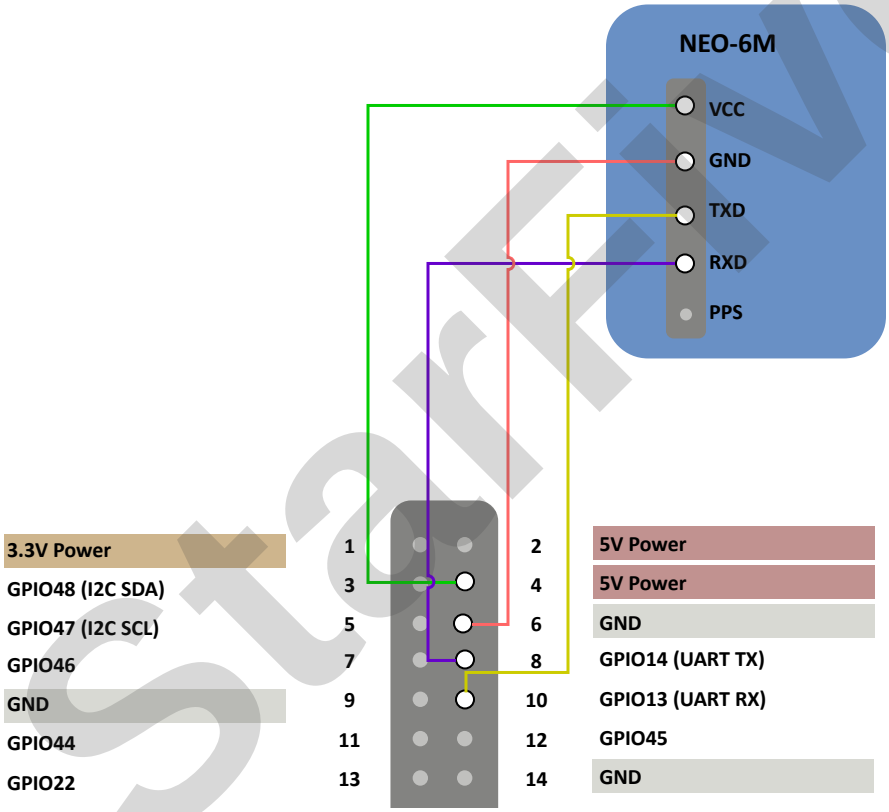
2.1.1. Hardware Setup

The following table and figure describe how to connect NEO-6M GPS to the 40-pin header:

Table 2-2 Connect NEO-6M GPS to the 40-Pin Header

NEO-6M	40-Pin GPIO Header	
	Pin Number	Pin Name
VCC	4	5V Power
GND	6	GND
TXD	10	UART RX
RXD	8	UART TX

Figure 2-1 Connect NEO-6M GPS to the 40-Pin Header



2.2. Preparing Software

Make sure the following procedures are performed:

1. Flash Fedora OS into a Micro-SD card as described in the *Flashing Fedora OS to a Micro-SD Card* section in the *VisionFive Single Board Computer Quick Start Guide*.
2. Log into the Fedora and make sure VisionFive is connected to the Internet. For detailed instructions, refer to the *Using SSH over Ethernet* or *Using a USB to Serial Converter* section in the *VisionFive Single Board Computer Quick Start Guide*.

3. Execute the `pip` command on VisionFive Fedora to install the `VisionFive.gpio` package:

```
sudo pip install VisionFive.gpio
```

Alternatively, you can execute the following command:

```
sudo pip3 install VisionFive.gpio
```

4. (Optional) If you copy the source code to the local directory under VisionFive Fedora, execute the following commands under the source code directory:

Tip:
The source code can be downloaded by clicking the following link: VisionFive.gpio.

```
sudo yum install python-devel python3-devel
sudo python setup.py install
```

Alternatively, you can execute the following command:

```
sudo python3 setup.py install
```

3. Running Demo Code

To run the demo code, perform the following on VisionFive Fedora:

1. Locate to the directory where the test code, `uart_gps_demo.py`, exists:

    a. Execute the following command to get the directory where `VisionFive.gpio` exists:

    ```
    pip show VisionFive.gpio
    ```

    Example Result:

    ```
    Location: /usr/local/lib64/python3.9/site-packages
    ```

    Note:

    The actual output depends on how the application is installed.

    b. Execute the following to enter the directory, for example, `/usr/local/lib64/python3.9/site-packages` as indicated in the previous step output:

    ```
    cd /usr/local/lib64/python3.9/site-packages
    ```

    c. Execute the following command to enter the `sample-code` directory:

    ```
    cd ./VisionFive/sample-code/
    ```

2. Execute the following command on your terminal before executing the demo code:

    ```
    sudo systemctl stop serial-getty@ttyS0.service
    ```

3. Under the `sample-code` directory, execute the following command to run the demo code:

    ```
    sudo python uart_gps_demo.py
    ```

    Alternatively, you can execute the following command:

    ```
    sudo python3 uart_gps_demo.py
    ```

    Result:

    If the GPS signal is weak, the terminal output is as the following:

    ```
    *****The GGA info is as follows: *****
     msg_id: $GPGGA
     NorS:
     EorW:
     pos_indi: 0
     total_Satellite: 00
    ```

```
!!!!!!Positioning is invalid!!!!!!
```

If the GPS signal is strong, the terminal output is as the following after a few seconds:

```
*****The GGA info is as follows: *****
 msg_id: $GPGGA
 utc time: 2:54:47.0
 utc time: 025447.00 (format: hhmmss.sss)
 latitude: 30 degree 33.29251 minute
 latitude: 3033.29251 (format: dddmm.mmmmm)
 NorS: N
 longitude: 104 degree 3.45523 minute
 longitude: 10403.45523 (format: dddmm.mmmmm)
 EorW: E
 pos_indi: 1
 total_Satellite: 08

*****The positioning type is 3D *****
The Satellite ID of channel {} : {}
                          ch1 : 14
                          ch2 : 01
                          ch3 : 03
                          ch4 : 06
                          ch5 : 30
                          ch6 : 21
                          ch7 : 19
                          ch8 : 17
```

4. Demo Source Code

The Python source code of this demo is provided for reference purpose only.

`uart_gps_demo.py`:

```
'''
Please make sure the NEO-6M is connected to the correct pins.
The following table describes how to connect NEO-6M to the 40-pin header
-----------------------------------------
Passive Buzzer___Pin Number_____Pin Name
    VCC             4           5 V Power
    GND             6             GND
    TXD             10          UART RX
    RXD             8           UART TX
-----------------------------------------
'''

import sys
import serial
import time

#Reference information of the GPGSA format.
'''
Example 1 (GPS only):

$GPGSA,M,3,17,02,30,04,05,10,09,06,31,12,,,1.2,0.8,0.9*35

Example 2 (Combined GPS and GLONASS):

$GNGSA,M,3,17,02,30,04,05,10,09,06,31,12,,,1.2,0.8,0.9*2B

$GNGSA,M,3,87,70,,,,,,,,,,1.2,0.8,0.9*2A

-------------------------------------------------------
SN      Field
                Description
                                Symbol
                                                Example
-------------------------------------------------------
1       $GPGSA
                Log header. For information about the log headers, see
 ASCII, Abbreviated ASCII or Binary.
                                N/A
                                                $GPGSA
2       mode MA
                Mode: 1 = Fix not available; 2 = 2D; 3 = 3D
                                x
```

```
                                          3
3       mode 123
                Latitude (DDmm.mm)
                                llll.ll
                                          5106.9847
4-15   prn
                PRN numbers of satellites used in solution (null for unused
 fields), total of 12 fields
                GPS = 1 to 32
                SBAS = 33 to 64 (add 87 for PRN number)
                GLO = 65 to 96
                                xx,xx,.....
                                          18,03,13,25,16,24,12,20,,,,

The detail info, please see
 https://docs.novatel.com/OEM7/Content/Logs/GPGSA.htm?tocpath=Commands%20%25
26%20Logs%7CLogs%7CGNSS%20Logs%7C_____63

'''

GPGSA_dict = {
"msg_id":  0,
"mode1":  1,
"mode2":  2,
"ch1":          3,
"ch2":          4,
"ch3":          5,
"ch4":          6,
"ch5":          7,
"ch6":          8,
"ch7":          9,
"ch8":          10,
"ch9":          11,
"ch10":         12,
"ch11":         13,
"ch12":         14,
}

#Reference information of the GPGGA format.
'''
Example 1 (GPS only):

$GPGSA,M,3,17,02,30,04,05,10,09,06,31,12,,,1.2,0.8,0.9*35

Example 2 (Combined GPS and GLONASS):

$GNGSA,M,3,17,02,30,04,05,10,09,06,31,12,,,1.2,0.8,0.9*2B

$GNGSA,M,3,87,70,,,,,,,,,,,1.2,0.8,0.9*2A
```

```
-------------------------------------------------------
SN   Field
               Description
                              Symbol
                                          Example
-------------------------------------------------------
1   $GPGGA
               Log header. For information about the log headers, see
 ASCII, Abbreviated ASCII or Binary.
                              N/A
                                          $GPGGA
2   utc
               UTC time status of position (hours/minutes/seconds/ decimal
 seconds)
                              hhmmss.ss
                                          202134.00
3   lat
               Latitude (DDmm.mm)
                              llll.ll
                                          5106.9847
4   lat dir
               Latitude direction (N = North, S = South)
                              a
                                          N
5   lon
               Latitude direction (N = North, S = South)
                              yyyyy.yy
                                          11402.2986
6   lon dir
               Longitude direction (E = East, W = West)
                              a
                                          W
7   quality
               refer to Table: GPS Quality Indicators
                              x
                                          1
8   # sats
               Number of satellites in use. May be different to the number
 in view
                              xx
                                          10
The detail info, please see
 https://docs.novatel.com/OEM7/Content/Logs/GPGGA.htm?tocpath=Commands%20%25
26%20Logs%7CLogs%7CGNSS%20Logs%7C_____59

'''
GPGGA_dict = {
"msg_id":   0,
"utc":           1,
"latitude":             2,
```

```
"NorS":                 3,
"longitude":            4,
"EorW":                 5,
"pos_indi":             6,
"total_Satellite":      7,
}

uart_port = "/dev/ttyS0"

def IsValidGpsinfo(gps):
    data = gps.readline()
    #Convert the data to string.
    msg_str = str(data, encoding="utf-8")
    #Split string with ",".
    #GPGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99*30
    msg_list = msg_str.split(",")

    #Parse the GPGSA message.
    if (msg_list[GPGSA_dict['msg_id']] == "$GPGSA"):
            print()
            #Check if the positioning is valid.
            if msg_list[GPGSA_dict['mode2']] == "1":
                print("!!!!!!Positioning is invalid!!!!!!")
            else:
                print("*****The positioning type is {}D
 *****".format(msg_list[GPGSA_dict['mode2']]))
                print("The Satellite ID of channel {} : {}")
                #Parse the channel information of the GPGSA message.
                for id in range(0, 12):
                    key_name = list(GPGSA_dict.keys())[id + 3]
                    value_id =  GPGSA_dict[key_name]
                    if  not (msg_list[value_id] == ''):
                        print("                              {} :
 {}".format(key_name, msg_list[value_id]))

    #Parse the GPGGA message.
    if msg_list[GPGGA_dict['msg_id']] == "$GPGGA":
        print()
        print("*****The GGA info is as follows: *****")
        for key, value in GPGGA_dict.items():
            #Parse the utc information.
            if key == "utc":
                utc_str =  msg_list[GPGGA_dict[key]]
                if not utc_str == '':
                    h = int(utc_str[0:2])
                    m = int(utc_str[2:4])
                    s = float(utc_str[4:])
                    print(" utc time: {}:{}:{}".format(h,m,s))
                    print(" {} time: {} (format: hhmmss.sss)".format(key,
 msg_list[GPGGA_dict[key]]))
```

```
            #Parse the latitude information.
            elif key == "latitude":
                lat_str = msg_list[GPGGA_dict[key]]
                if not lat_str == '':
                    Len = len(lat_str.split(".")[0])
                    d = int(lat_str[0:Len-2])
                    m = float(lat_str[Len-2:])
                    print(" latitude: {} degree {} minute".format(d, m))
                    print(" {}: {} (format: dddmm.mmmmm)".format(key,
 msg_list[GPGGA_dict[key]]))
            #Parse the longitude information.
            elif key == "longitude":
                lon_str = msg_list[GPGGA_dict[key]]
                if not lon_str == '':
                    Len = len(lon_str.split(".")[0])
                    d = int(lon_str[0:Len-2])
                    m = float(lon_str[Len-2:])
                    print(" longitude: {} degree {} minute".format(d, m))
                    print(" {}: {} (format: dddmm.mmmmm)".format(key,
 msg_list[GPGGA_dict[key]]))
            else:
                print(" {}: {}".format(key, msg_list[GPGGA_dict[key]]))

def main():
    gps = serial.Serial(uart_port, baudrate=9600, timeout=0.5)
    while True:
        IsValidGpsinfo(gps)
        time.sleep(1)

    gps.close()

if __name__ == "__main__":
    sys.exit(main())
```