



StarFive  
赛昉科技

## 使用昉·星光 2的UART读取GPS数据

Python语言版本

应用说明

版本： 1.2

日期： 2025/08/07

Doc ID: VisionFive 2-ANCH-007

# 法律声明

阅读本文件前的重要法律告知。

## 版权注释

版权 ©上海赛昉半导体科技有限公司，2025。版权所有。

本文档中的说明均基于“视为正确”提供，可能包含部分错误。内容可能因产品开发而定期更新或修订。上海赛昉半导体科技有限公司（以下简称“赛昉科技”）保留对本协议中的任何内容进行更改的权利，恕不另行通知。

赛昉科技明确否认任何形式的担保、解释和条件，无论是明示的还是默示的，包括但不限于适销性、特定用途适用性和非侵权的担保或条件。

赛昉科技无需承担因应用或使用任何产品或电路而产生的任何责任，并明确表示无需承担任何及所有连带责任，包括但不限于间接、偶然、特殊、惩戒性或由此造成的损害。

本文件中的所有材料受版权保护，为赛昉科技所有。不得以任何方式修改、编辑或断章取义本文件中的说明，本文件或其任何部分仅限用于内部使用或教育培训。使用文件中包含的说明，所产生的风险由您自行承担。赛昉科技授权复制本文件，前提是您保留原始材料中包含的所有版权声明和其他相关声明，并严格遵守此类条款。本版权许可不构成对产品或服务的许可。

## 联系我们：

地址：中国（上海）自由贸易试验区盛夏路61弄张润大厦2号电梯楼层5层（实际楼层4层）06室

网站：<http://www.starfivetech.com>

邮箱：[sales@starfivetech.com](mailto:sales@starfivetech.com)（销售） [support@starfivetech.com](mailto:support@starfivetech.com)（支持）

# 前言

关于本指南和技术支持信息

## 关于本手册

本应用指南提供了使用昉·星光 2的UART，通过Python示例程序读取GPS数据的步骤。






## 修订历史

表 0-1 修订历史

版本	发布说明	修订
1.2	2025/08/07	更新了以下章节： <ul style="list-style-type: none"><li>• <a href="#">运行环境要求 (第 8页)</a></li><li>• <a href="#">准备软件 (第 9页)</a></li><li>• <a href="#">执行演示代码 (第 12页)</a></li></ul>
1.1	2023/06/08	<ul style="list-style-type: none"><li>• 在<a href="#">40-Pin GPIO Header定义 (第 7页)</a>增加注释。</li><li>• 在<a href="#">准备软件 (第 9页)</a>中更新安装方式。</li><li>• 新增<a href="#">资源下载 (第 18页)</a>和<a href="#">立即购买 (第 19页)</a>章节。</li></ul>
1.0	2022/12/15	首次发布。

## 注释和注意事项

本指南中可能会出现以下注释和注意事项：

-  **提示：**  
建议如何在某个主题或步骤中应用信息。
-  **注：**  
解释某个特例或阐释一个重要的点。
-  **重要：**  
指出与某个主题或步骤有关的重要信息。
-  **警告：**  
表明某个操作或步骤可能会导致数据丢失、安全问题或性能问题。
-  **警告：**  
表明某个操作或步骤可能导致物理伤害或硬件损坏。

---

# 目录

表格清单.....	5
插图清单.....	6
法律声明.....	2
前言.....	3
<b>1. 产品简介.....</b>	<b>7</b>
1.1. 40-Pin GPIO Header定义.....	7
<b>2. 准备.....</b>	<b>8</b>
2.1. 运行环境要求.....	8
2.2. 准备硬件.....	8
2.2.1. 连接硬件.....	8
2.3. 准备软件.....	9
<b>3. 执行演示代码.....</b>	<b>12</b>
<b>4. 演示源代码.....</b>	<b>14</b>
<b>5. 资源下载.....</b>	<b>18</b>
<b>6. 立即购买.....</b>	<b>19</b>



StarFive  
赛昉科技

# 表格清单

表 0-1 修订历史.....	3
表 2-1 硬件准备.....	8
表 2-2 将NEO-6M GPS连接到40-Pin GPIO Header上.....	8



## 插图清单

图 1-1 40-Pin GPIO Header定义.....	7
图 2-1 将NEO-6M GPS连接到40-Pin GPIO Header上.....	9



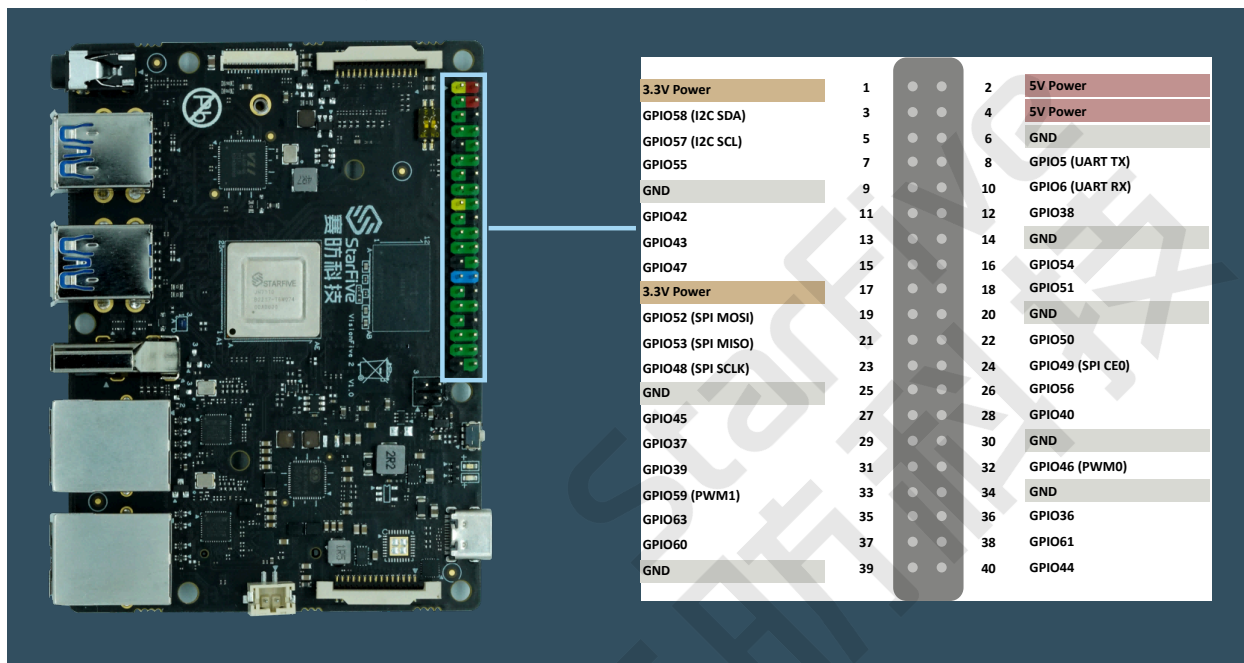
# 1. 产品简介

本应用指南提供了使用昉·星光 2的UART，通过Python示例程序读取GPS数据的步骤。

## 1.1. 40-Pin GPIO Header定义

下图显示了40-pin GPIO Header的位置：

图 1-1 40-Pin GPIO Header定义



注：

功能复用pin脚已初始化，不可作为通用GPIO使用。

## 2. 准备

在执行演示程序之前，务必确认已准备好以下项目：

### 2.1. 运行环境要求

该演示运行环境要求如下：

- Linux内核版本：Linux 6.6
- 操作系统：Debian 13
- 硬件版本：昉·星光 2
- SoC：昉·惊鸿-7110

### 2.2. 准备硬件

在执行演示程序之前，请务必准备以下硬件：

表 2-1 硬件准备

类型	M/O*	项目	注释
通用	M	昉·星光 2 单板计算机	-
通用	M	<ul style="list-style-type: none"><li>• 容量不低于32 GB的Micro-SD卡</li><li>• Micro-SD卡读卡器</li><li>• 计算机 (Windows/Mac OS/Linux)</li><li>• USB转串口转换器 (3.3 V I/O, 带线)</li><li>• 以太网电缆</li><li>• 电源适配器 (5 V/ 3 A)</li><li>• USB Type-C数据线</li></ul>	上述项目用于将Debian OS烧录到Micro-SD上。
UART演示	M	<ul style="list-style-type: none"><li>• NEO-6M GPS</li><li>• 4根杜邦线 (母对母)</li><li>• 外部天线 (可选)</li></ul>	天线用于改善GPS信号接收。



注：

\*: M：必须。O：可选

#### 2.2.1. 连接硬件

以下表格和图片描述了如何将NEO-6M GPS连接到40-Pin GPIO Header上：

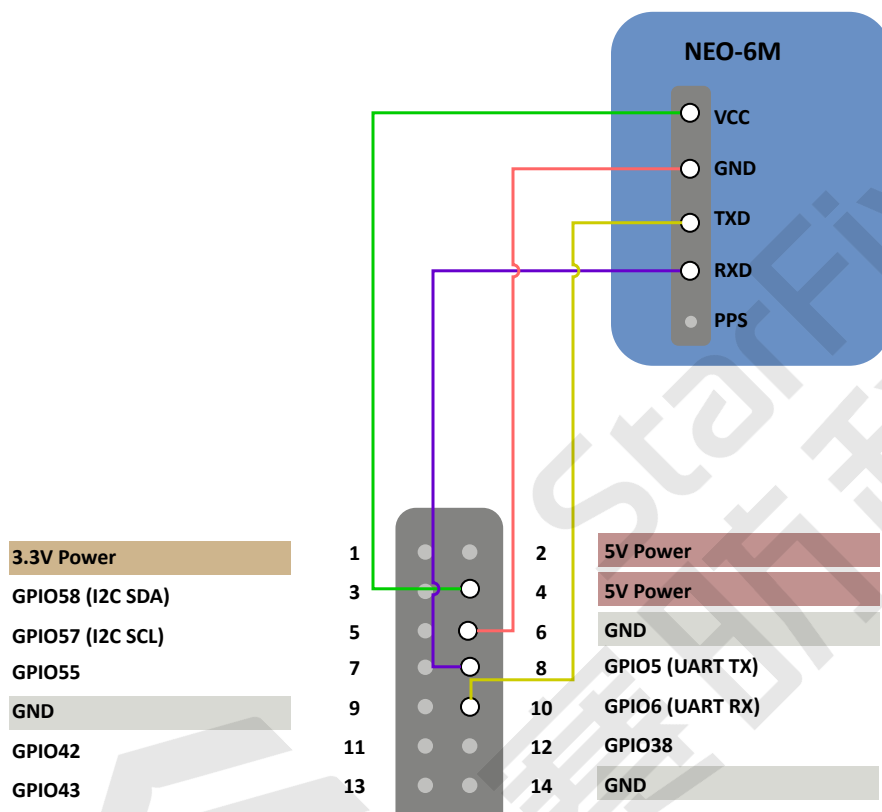
表 2-2 将NEO-6M GPS连接到40-Pin GPIO Header上

NEO-6M	40-Pin GPIO Header	
	Pin Number	Pin Name
VCC	4	5V 电压
GND	6	GND

表 2-2 将NEO-6M GPS连接到40-Pin GPIO Header上 (续)

NEO-6M	40-Pin GPIO Header	
	Pin Number	Pin Name
TXD	10	GPIO6 (UART RX)
RXD	8	GPIO5 (UART TX)

图 2-1 将NEO-6M GPS连接到40-Pin GPIO Header上



## 2.3. 准备软件

确认按照以下步骤进行操作：



注：

该Python应用VisionFive.gpio适用于昉·星光单板计算机、昉·星光 2和昉·惊鸿-7110 EVB。

1. 按照[《昉·星光 2单板计算机快速参考手册》](#)中的“将OS烧录到Micro-SD”章节，将Debian OS烧录到Micro-SD卡上。
2. 登录Debian并确保昉·星光 2已联网。有关详细说明，请参阅[《昉·星光 2单板计算机快速参考手册》](#)中“通过以太网使用SSH”或“使用USB转串口转换器”章节。
3. 在Debian上扩展分区，请参见[《昉·星光 2单板计算机快速参考手册》](#)中“扩展分区”章节。
4. 在Debian上执行以下命令以安装并创建Python3虚拟环境：

```
sudo apt install python3-venv
python3 -m venv myvenv
```



**注：**  
可自定义 "myvenv" 名称。

5. 在昉·星光 2 Debian上执行pip命令，以安装VisionFive.gpio包：



**注：**  
由于pypi.org官网尚不支持上传RISC-V平台的whl安装包，不能直接执行`python3 -m pip install VisionFive.gpio`命令在线安装，因此请按照以下步骤安装VisionFive.gpio包。

a. 执行以下命令，在新创建的虚拟环境中安装依赖包：

```
sudo apt install libxml2-dev libxslt-dev
source ./myvenv/bin/activate
python3 -m pip install requests wget bs4
```

b. 执行以下命令，运行安装脚本Install\_VisionFive\_gpio.py：

```
python3 Install_VisionFive_gpio.py
```

安装脚本代码如下：

```
import requests
import wget
import sys
import os
from bs4 import BeautifulSoup

def parse_data(link_addr, class_type, key_str):
    req = requests.get(url=link_addr)
    req.encoding = "utf-8"
    html = req.text
    soup = BeautifulSoup(req.text, features="html.parser")
    package_version = soup.find(class_type, class_=key_str)
    dd = package_version.text.strip()
    data = dd.split()
    return data

def parse_link(link_addr, class_type, key_str):
    version_list = []
    req = requests.get(url=link_addr)
    req.encoding = "utf-8"
    html = req.text
    soup = BeautifulSoup(req.text, features="html.parser")
    search_data = soup.find_all(class_type, class_=key_str)
    for i in range(0, len(search_data)):
        search_data[i] = search_data[i].find("a").get("href")
        version_list.append(search_data[i].split("cp")[-1].split("-")[0])

    python_version = sys.version
    python_version = python_version.split(".")[0] + python_version.split(".")[1]

    for i in range(0, len(search_data)):
        if python_version == version_list[i]:
            return search_data[i]

    return search_data[0]

def get_dl_addr_page():
    link_address = "https://pypi.org/project/VisionFive.gpio/#history"
    key_str = "release version"
    class_key = "p"
    data_get = parse_data(link_address, class_key, key_str)
    latest_version = data_get[0]
    dl_addr_page = "https://pypi.org/project/VisionFive.gpio/{}/#files".format(latest_version)
    return dl_addr_page

def get_dl_addr_of_latest_version(link_addr):
    key_str = "card file card"
```



```
class_key = "div"
addr_get = parse_link(link_addr, class_key, key_str)

return addr_get

def main():
    dl_addr_p = get_dl_addr_page()
    whl_dl_addr = get_dl_addr_of_latest_version(dl_addr_p)

    whl_name = whl_dl_addr.split("/")[-1]
    whl_name_suffix = os.path.splitext(whl_name)[-1]
    whl_name_prefix = os.path.splitext(whl_name)[0]
    whl_name_prefix_no_platform = whl_name_prefix[0: len(whl_name_prefix) - 3]
    new_platform = "linux_riscv64"

    rename_whl_name = "{}{}{}".format(whl_name_prefix_no_platform, new_platform,
whl_name_suffix)

    wget.download(whl_dl_addr, out=rename_whl_name)

    os.system("pip install " + rename_whl_name)
    os.system("rm -rf " + rename_whl_name)

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

c. (可选) 退出 Python3 虚拟环境。

```
deactivate
```

### 3. 执行演示代码

执行以下操作，以在昉·星光 2的Debian系统上运行演示代码：

1. 找到测试代码`uart_gps_demo.py`所在的目录：

- a. 进入 Python3 虚拟环境：

```
source ./myvenv/bin/activate
```

- b. 执行以下命令安装依赖：

```
python3 -m pip install pillow
```

- c. 执行以下命令以获取`VisionFive.gpio`所在的目录：

```
python3 -m pip show VisionFive.gpio
```

**示例结果：**

```
Location: /home/user/myvenv/lib/python3.11/site-packages
```



**注：**

实际输出取决于应用的安装方式。

- d. 执行以下命令进入目录，例如上一步所示的 `/home/user/myvenv/lib/python3.11/site-packages`：

```
cd /home/user/myvenv/lib/python3.11/site-packages
```

- e. 执行以下命令进入`sample-code`目录：

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

2. 在执行演示代码前，在您的终端上执行以下命令：

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

3. 在`sample-code`目录下，执行以下命令以运行演示代码：

```
sudo python uart_gps_demo.py
```

或者，您也可以执行以下命令：

```
sudo python3 uart_gps_demo.py
```

**结果：**

如果GPS信号弱，终端输出如下：

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

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

如果GPS信号强，几秒后终端输出如下：

```
*****The GGA info is as follows: *****
msg_id: print(" 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. (可选) 退出 Python3 虚拟环境。

```
deactivate
```

## 4. 演示源代码

本演示中的资源代码仅作为参考。

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)
        1111.11
        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%2526%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 GPFGA 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)
      1111.11
      5106.9847
4   lat dir
      Latitude direction (N = North, S = South)
      a
      N
5   lon
      Latitude direction (N = North, S = South)
      YYYY.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%2526%20Logs%7CLogs%7CGNSS%20Logs%7C\_\_\_\_\_59
'''
GPFGA_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 {} : {}".format(key_name, msg_list[value_id]))
            #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: hhhmss.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())
```



---

## 5. 资源下载

点击本栏找到所有的代码下载资源。

本页包括所有赛昉科技提供的代码下载资源。

- [RVspace Wiki](#)
- [应用中心](#)
- [文档中心](#)
- [技术论坛](#)
- [昉·星光 2 GitHub代码仓](#)
- [昉·星光 2 Debian操作系统下载](#)
- [代码下载 \(赛昉科技官方GitHub页面\)](#)
- [所有开源技术文档](#)



StarFive  
赛昉科技

---

## 6. 立即购买

点击本栏获取在线购买链接和配件购买链接。

### 购买单板计算机

点击以下页面，您可以找到所在地区的经销商，或覆盖全球的销售渠道，以购买昉·星光 2 单板计算机。

- [购买昉·星光 2 开发板](#)

### 购买配件

点击以下页面，您可以找到所有昉·星光 2 单板计算机已验证适配的配件及其购买链接。

- [购买配件](#)

