



StarFive  
赛昉科技

# Using VisionFive UART to Read GPS Data Application Note

Version: V1.1

Date: 2021-12-29

## PROPRIETARY NOTICE

Copyright © Shanghai StarFive Technology Co., Ltd., 2018-2022. All rights reserved.

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.

All material appearing in this document is protected by copyright and is the property of StarFive. You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. StarFive authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services.

### Shanghai StarFive Technology Co., Ltd.

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

Website: [www.starfivetech.com](http://www.starfivetech.com)

Email: [sales@starfivetech.com](mailto:sales@starfivetech.com) (sales)  
[support@starfivetech.com](mailto:support@starfivetech.com) (support)

# About This Manual

## Introduction

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

## Revision History

Version	Released	Revision
V1.0	2021-12-15	Preliminary release.
V1.1	2021-12-29	<ul style="list-style-type: none"><li>• Updated the Makefile content format.</li><li>• Updated the Makefile description.</li><li>• Added description for the <code>rsync</code> command.</li><li>• Added description for <code>&lt;User_Name&gt;</code>.</li></ul>

## Table of Contents

---

<b>About This Manual .....</b>	<b>ii</b>
<b>1 Preparation .....</b>	<b>4</b>
1.1 Preparing Hardware .....	4
1.1.1 Hardware Setup .....	4
1.1.2 Hardware Setup .....	4
1.2 Preparing Software .....	6
1.2.1 Software Environment .....	6
1.2.2 Preparing Toolchain .....	6
1.2.3 Preparing UART GPIO .....	6
<b>2 Running Demo Codes .....</b>	<b>7</b>

StarFive

# 1 Preparation

Before executing the demo program, make sure you have prepared the following:

## 1.1 Preparing Hardware

Table 1-1 Hardware Preparation

Type	M/O	Item	Notes
General	M	A Single Board Computer	The following boards are applicable: <ul style="list-style-type: none"><li>• StarLight</li><li>• VisionFive</li></ul>
General	M	<ul style="list-style-type: none"><li>• 16GB (or more) micro-SD card</li><li>• micro-SD card reader</li><li>• Computer (PC/Mac/Linux)</li><li>• USB to serial converter (3.3 V I/O)</li><li>• Ethernet cable</li><li>• Power adapter (5 V / 3 A)</li><li>• USB Type-C Cable</li></ul>	These items are used for flashing Fedora OS into a micro-SD card.
UART Demo	M	<ul style="list-style-type: none"><li>• GNSS HAT</li><li>• Dupont Line</li></ul>	This is a GNSS HAT based on MAX-7Q, which supports positioning systems including GPS, GLONASS, QZSS, and SBAS. It features accurate and fast positioning with minor drifting, low power consumption, outstanding ability for anti-spoofing and anti-jamming, and so on. For detailed specifications, refer to <a href="#">MAX-7Q GNSS HAT</a> .

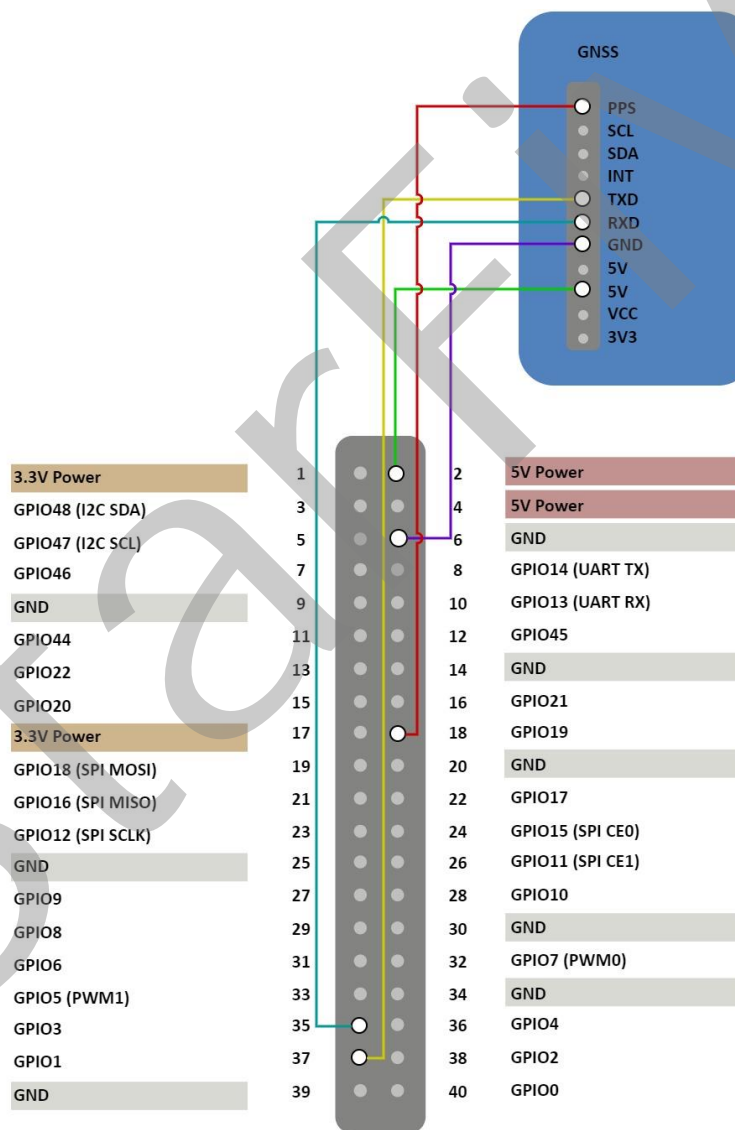
\*M/O: M (Mandatory)/ O (Optional)

### 1.1.2 Hardware Setup

The following table and figure describe how to connect GNSS HAT to the 40-pin header:

**Table 1-2 onnect GNSS HAT to the 40-Pin Header**

GNSS HAT	Pin Number
PPS	18
TXD	37
RXD	35
GND	6
5V	2



**Figure 1-1 Connect GNSS HAT to the 40-Pin Header**

## 1.2 Preparing Software

### 1.2.1 Software Environment

- PC: Ubuntu 20.04
- RISC-V Platform: Linux 5.16.0

### 1.2.2 Preparing Toolchain

Install the tool to compile. The following is an example to install:

```
sudo apt-get install gcc-riscv64-linux-gnu
```

#### Information:

This step can be skipped if the tool has been installed.

After successful installation, check the version by running: `linus@starfive$ riscv64-linux-gnu-gcc -v`. The following is the example output:

```
Thread model: posix
gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.04)
```

Figure 1-2 Example Output

### 1.2.3 Preparing UART GPIO

Make sure the following procedures are performed:

1. Flash Fedora OS into a Micro-SD card and compile and replace dtb files as described in the *Preparing Software* section in *StarFive 40-Pin GPIO Header User Guide*.
2. Configure the GPIO pin as UART by setting the dts file as described in the *Configuring UART GPIO* section in the *StarFive 40-Pin GPIO Header User Guide*.

#### Information:

You can configure the unoccupied pins as UART. The following is an example table for the mapping:

Table - UART and Pin Name Mapping

UART	GPIO (Pin Name)
Uart1	<ul style="list-style-type: none"> <li>• GPIO3</li> <li>• GPIO1</li> </ul>
Uart2	<ul style="list-style-type: none"> <li>• GPIO2</li> <li>• GPIO0</li> </ul>

## 2 Running Demo Codes

To run the demo codes, perform the following:

**Step 1** Create test-gps file under app directory to save test file.

**Step 2** Download the source code from: [test-gps.c](https://test-gps.c).

**Step 3** Execute the following to create Makefile:

```
touch Makefile
```

**Step 4** Copy the following to the Makefile, save and exit:

```
EXEC = test-gps
OBJS = test-gps.o

#CROSS = riscv64-unknown-elf-
CROSS = riscv64-linux-gnu-
CC = $(CROSS)gcc
STRIP = $(CROSS)strip
CFLAGS = -O2

all: clean $(EXEC)

$(EXEC):$(OBJS)
    $(CC) $(CFLAGS) -o $@ $(OBJS)
    $(STRIP) $@

clean:
    -rm -f $(EXEC) *.o
```

**Step 5** Execute make to generate the executable test-gps file.

**Step 6** Execute file command to check if it is a RISC-V file. The following is an example output:

```
linus@starfive:~/work/starlingt_puls/app/docs$ file test-gps
test-gps: ELF 64-bit LSB shared object, UCB RISC-V, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-riscv64-lp64d.so.1, BuildID[sha1]=6ad296f8420149b81bdfa344ba198a3bde8b74c, for GNU/Linux 4.15.0, stripped
linus@starfive:~/work/starlingt_puls/app/docs$
```

Figure 2-1 Example Output

**Step 7** Execute the following command in Ubuntu to upload the executable file test-gps to your desired directory of the board, for example, test:

```
rsync ./test-gps <User_Name>@<Board_IP_Address>:/home/riscv/test
```

**Information:**



- <User\_Name>: Your user name of the board. For example, riscv.
- <Board\_IP\_Address>: The board IP address. For example, 192.168.92.133.

**Example:**

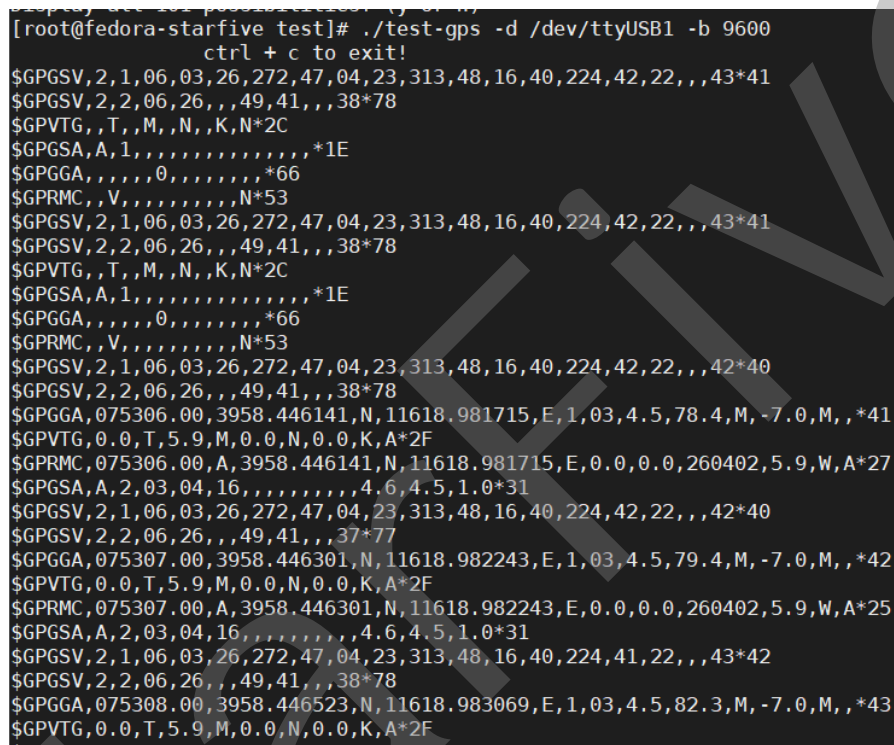
```
rsync ./test-gps riscv@192.168.92.133:/home/riscv/test
```

**Step 8** Execute the following command on VisionFive to run the demo codes:

```
./test-gps -d /dev/ttyUSB1 -b 9600
```

**Result:**

The following output indicates the execution is successful:



```
[root@fedora-starfive test]# ./test-gps -d /dev/ttyUSB1 -b 9600
ctrl + c to exit!
$GPGSV,2,1,06,03,26,272,47,04,23,313,48,16,40,224,42,22,,43*41
$GPGSV,2,2,06,26,,49,41,,38*78
$GPVTG,,T,,M,,N,,K,N*2C
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPGGA,,,,,0,,,,,,,,,*66
$GPRMC,,V,,,,,,,,,N*53
$GPGSV,2,1,06,03,26,272,47,04,23,313,48,16,40,224,42,22,,43*41
$GPGSV,2,2,06,26,,49,41,,38*78
$GPVTG,,T,,M,,N,,K,N*2C
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPGGA,,,,,0,,,,,,,,,*66
$GPRMC,,V,,,,,,,,,N*53
$GPGSV,2,1,06,03,26,272,47,04,23,313,48,16,40,224,42,22,,42*40
$GPGSV,2,2,06,26,,49,41,,38*78
$GPGGA,075306.00,3958.446141,N,11618.981715,E,1,03,4.5,78.4,M,-7.0,M,,*41
$GPVTG,0.0,T,5.9,M,0.0,N,0.0,K,A*2F
$GPRMC,075306.00,A,3958.446141,N,11618.981715,E,0.0,0.0,260402,5.9,W,A*27
$GPGSA,A,2,03,04,16,,,,,,,,,4.6,4.5,1.0*31
$GPGSV,2,1,06,03,26,272,47,04,23,313,48,16,40,224,42,22,,42*40
$GPGSV,2,2,06,26,,49,41,,37*77
$GPGGA,075307.00,3958.446301,N,11618.982243,E,1,03,4.5,79.4,M,-7.0,M,,*42
$GPVTG,0.0,T,5.9,M,0.0,N,0.0,K,A*2F
$GPRMC,075307.00,A,3958.446301,N,11618.982243,E,0.0,0.0,260402,5.9,W,A*25
$GPGSA,A,2,03,04,16,,,,,,,,,4.6,4.5,1.0*31
$GPGSV,2,1,06,03,26,272,47,04,23,313,48,16,40,224,41,22,,43*42
$GPGSV,2,2,06,26,,49,41,,38*78
$GPGGA,075308.00,3958.446523,N,11618.983069,E,1,03,4.5,82.3,M,-7.0,M,,*43
$GPVTG,0.0,T,5.9,M,0.0,N,0.0,K,A*2F
```

**Figure 2-2 Example Output**

Refer to <http://aprs.gids.nl/nmea/> for how to analyze the GPS data.