RTStreamHub User Manual

RTStreamHub: A headless GNSS data streaming hub with NTRIP TLS support

written by Runzhi Zhang, Rui Tu, Xiaochun Lu



**National Time Service Center,**

**Chinese Academy of Sciences**

Update: 2025.03.10

Contents

[1 Introduction 2](#_Toc192597118)

[2 Download and Functionality 3](#_Toc192597119)

[3 Configuration and Execution 5](#_Toc192597120)

[3.1 Preparation and Configuration 5](#_Toc192597121)

[3.2 Execution 5](#_Toc192597122)

[4 Compile and Package 6](#_Toc192597123)

[4.1 Compile 6](#_Toc192597124)

[4.2 Package 7](#_Toc192597125)

[Contact Us 8](#_Toc192597126)

# 1 Introduction

The forwarding, caching, and transmission of real-time data streams are crucial components of GNSS data processing. Various communication protocols, such as the transmission control protocol (TCP), the networked transport of RTCM via internet protocol (NTRIP), and serial ports, play a significant role in data reception and transmission. Receiver data is transmitted to the server through a data transfer unit (DTU) via the TCP. The server then forwards or pushes the data stream to an NTRIP caster, or alternatively, real-time precise products are delivered to users by organizations such as the International GNSS Service (IGS) through the NTRIP. Particularly, the adoption of NTRIP TLS (transport layer security) enhances data confidentiality for users. However, the reception of NTRIP TLS data on the user side involves more complex procedures.

Therefore, RTStreamHub, an open-source, headless C/C++ software that operates on Linux servers, facilitates the forwarding, caching, and transmission of real-time data streams. Built on RTKLIB, RTStreamHub has been further upgraded to support NTRIP, with enhanced compatibility for NTRIP TLS and the use of port 443 for TCP encryption, ensuring secure data reception. This development enables users to conveniently receive real-time satellite products, such as the high accuracy service (HAS) provided by European Galileo.

RTStreamHub provides an example for forwarding an incoming data stream. By running RTStreamHub on a Linux server, users can easily implement data stream forwarding. Furthermore, RTStreamHub can also be embedded into the server and client sides of RTK (real-time kinematic) and PPP (precise point positioning) programs to enable the broadcasting of data streams.

The remaining chapters are arranged as follows: Chapter 2 introduces the download and functionality of RTStreamHub. Chapter 3 covers the configuration and execution of RTStreamHub. Chapter 4 discusses the compilation and debugging of RTStreamHub.

# 2 Download and Functionality

RTStreamHub is supported for execution on Linux systems, and the OpenSSL library must be supported as well. The source code, executable files, and manual of RTStreamHub are all contained within the **RTStreamHub** folder. The contents included are as follow:

图形用户界面

AI 生成的内容可能不正确。

The source code is contained within the **src** folder, while the executable files are in the **bin** folder. A Qt project is included in the **RTStreamHub** folder, and the configuration files are stored in the **config** folder. Configuration files are read using relative paths, so it is essential to ensure that the **debug** folder used for debugging and the **bin** folder for execution, along with the **config** folder, are located within the same parent directory.

RTStreamHub is specifically designed for data forwarding, including the conversion of data transmission methods such as serial port, TCP, and NTRIP. It also supports data collection into files. The specific functionalities are depicted in the diagram below.



The data collected by the GNSS receiver is aggregated and then transmitted to the server via serial port or DTU. RTStreamHub, executed on a Linux server, forwards the collected data. RTStreamHub supports inputs serial port, TCP client, TCP server, NTRIP caster, and NTRIP client with or without TLS, and forwards outputs File, TCP server, TCP client, serial port and NTRIP caster.

# 3 Configuration and Execution

## 3.1 Preparation and Configuration

RTStreamHub is executed on the Linux operating system, and the OpenSSL library must be installed on the Linux system. To check if OpenSSL is installed on the system, run the following command:

openssl version

If OpenSSL is not installed, you can install it with the following command (Ubuntu):

sudo apt update

sudo apt install openssl

Additionally, the Qt libraries are required, and the program is compiled with Qt version 5.12 on Ubuntu 18.04. On Ubuntu, you can install Qt 5.12 or later using the following command:

sudo apt update

sudo apt install qt5-qmake qtbase5-dev

The configuration of RTStreamHub is simple. The **set/input** option in strsvr.ini which is in the **config** folder defines the input for the program, with available choices being 0 (serial port), 1 (TCP client), 2 (TCP server), 3 (NTRIP client), 4 (NTRIP caster), and 5 (File). The program provides three data forwarding streams, which can be defined via **set/output1**, **set/output2**, and **set/output3**. The available options include: 0 (no output), 1 (serial port), 2 (TCP client), 3 (TCP server), 4 (NTRIP server), 5 (NTRIP caster), and 6 (file).

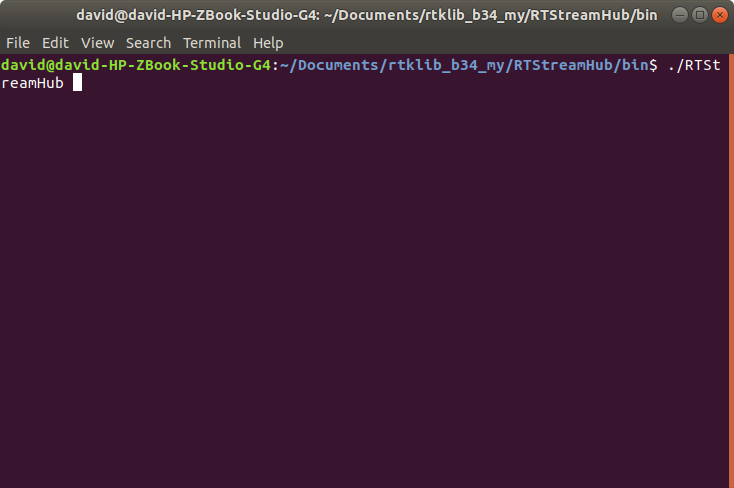


The parameters for the data streams are configured through **path**, with each data stream corresponding to four parameters **path\_A\_B**, where A refers to the data stream, and B refers to the protocol type. The available protocol types include: 0 (serial port), 1 (TCP and NTRIP), and 2 (file). For serial port, the format is *Port:Bitrate:ByteSize:Parity:StopBit:FlowControl*; for TCP and NTRIP, the format is *Userid:Password@ip:port/mountpoint*; and for files, the format is *AbsolutePath::S=24*. The NTRIP client, with or without TLS, is distinguished by whether the port number is 443.

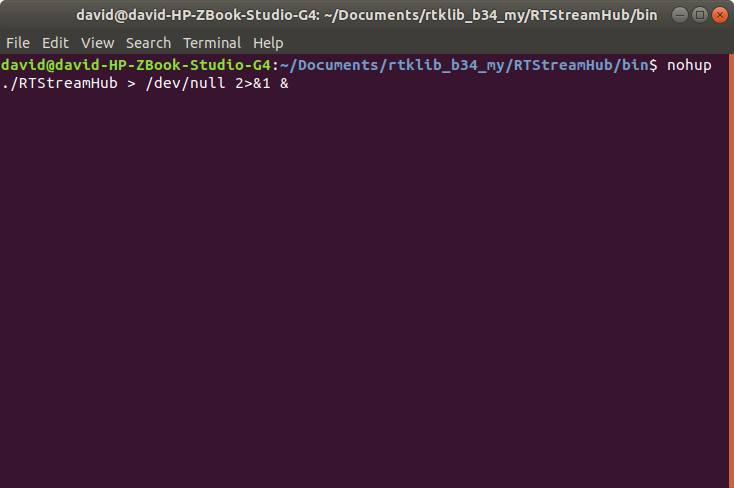
The program’s log can be configured using **set/tracefile** and **set/tracelevel**. The **tracelevel** option supports values from 0 to 5, with the level increasing in detail.

## 3.2 Execution

The program executable is in the **bin** folder. It is important to note that the program configuration file is read using relative paths. Therefore, the **bin** and **config** folders must be in the same directory and at the same hierarchical level. The program is executed as follows (Ubuntu 18.04-x64 tested):



Run in the background:



nohup ./ RTStreamHub > /dev/null 2>&1 &

# 4 Compile and Package

## 4.1 Compile

You can compile the source code of this project using QT.

1. Preparation

Before proceeding with RTStreamHub project, ensure you first install QT from the official website (<www.qt.io>). Once QT installation is completed, verify that the integrated development environment is correctly configured. RTStreamHub has been tested on Linux with “GCC 5.12.0 64-bit” environments.

1. Open project

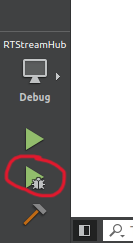
Open the file “RTStreamHub\**RTStreamHub.pro**” with QT as shown below:

电脑萤幕的截图

AI 生成的内容可能不正确。

1. Run

Debug and run this case by clicking on the *red circle* as shown below:



## 4.2 Package

Users can package the executable program with QT as follow:

1. Run in Release mode in QT creator to generate executable file:

图片包含 应用程序

AI 生成的内容可能不正确。

1. Package program

* Linux users (GCC 5.12.0 64-bit):

All the relevant library files must be included, otherwise the program cannot run successfully (Command “*ldd RTStreamHub*” can list all the library files of **RTStreamHub** in “Terminal” for checking), as shown below:

文本

AI 生成的内容可能不正确。

# Contact Us

We welcome all users to share and exchange technical details, as well as provide improvement suggestions. Here are our contact details:

QQ Groups: 651080705 (HASPPP)

GitHub: ZhangRunzhi20 (NTSC\_zrz)

Online HAS dataset: https://github.com/ZhangRunzhi20/

Emails: [zhangrunzhi@ntsc.ac.cn](mailto:zhangrunzhi@ntsc.ac.cn); [turui-2004@126.com](mailto:turui-2004@126.com); [luxc@ntsc.ac.cn](mailto:luxc@ntsc.ac.cn)

Address: No. 3 Shuyuan East Road, Xi'an City, Shaanxi Province, China

Postal code: 710600