

# L89 R2.0&LC29H&LC79H

## Firmware Upgrade Guide

**GNSS Module Series**

Version: 1.2

Date: 2022-04-12

Status: Released



At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:

**Quectel Wireless Solutions Co., Ltd.**

Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: [info@quectel.com](mailto:info@quectel.com)

**Or our local offices. For more information, please visit:**

<http://www.quectel.com/support/sales.htm>.

**For technical support, or to report documentation errors, please visit:**

<http://www.quectel.com/support/technical.htm>.

Or email us at: [support@quectel.com](mailto:support@quectel.com).

## Legal Notices

We offer information as a service to you. The provided information is based on your requirements and we make every effort to ensure its quality. You agree that you are responsible for using independent analysis and evaluation in designing intended products, and we provide reference designs for illustrative purposes only. Before using any hardware, software or service guided by this document, please read this notice carefully. Even though we employ commercially reasonable efforts to provide the best possible experience, you hereby acknowledge and agree that this document and related services hereunder are provided to you on an “as available” basis. We may revise or restate this document from time to time at our sole discretion without any prior notice to you.

## Use and Disclosure Restrictions

### License Agreements

Documents and information provided by us shall be kept confidential, unless specific permission is granted. They shall not be accessed or used for any purpose except as expressly provided herein.

### Copyright

Our and third-party products hereunder may contain copyrighted material. Such copyrighted material shall not be copied, reproduced, distributed, merged, published, translated, or modified without prior written consent. We and the third party have exclusive rights over copyrighted material. No license shall be granted or conveyed under any patents, copyrights, trademarks, or service mark rights. To avoid ambiguities, purchasing in any form cannot be deemed as granting a license other than the normal non-exclusive, royalty-free license to use the material. We reserve the right to take legal action for noncompliance with abovementioned requirements, unauthorized use, or other illegal or malicious use of the material.

## Trademarks

Except as otherwise set forth herein, nothing in this document shall be construed as conferring any rights to use any trademark, trade name or name, abbreviation, or counterfeit product thereof owned by Quectel or any third party in advertising, publicity, or other aspects.

## Third-Party Rights

This document may refer to hardware, software and/or documentation owned by one or more third parties (“third-party materials”). Use of such third-party materials shall be governed by all restrictions and obligations applicable thereto.

We make no warranty or representation, either express or implied, regarding the third-party materials, including but not limited to any implied or statutory, warranties of merchantability or fitness for a particular purpose, quiet enjoyment, system integration, information accuracy, and non-infringement of any third-party intellectual property rights with regard to the licensed technology or use thereof. Nothing herein constitutes a representation or warranty by us to either develop, enhance, modify, distribute, market, sell, offer for sale, or otherwise maintain production of any our products or any other hardware, software, device, tool, information, or product. We moreover disclaim any and all warranties arising from the course of dealing or usage of trade.

## Privacy Policy

To implement module functionality, certain device data are uploaded to Quectel’s or third-party’s servers, including carriers, chipset suppliers or customer-designated servers. Quectel, strictly abiding by the relevant laws and regulations, shall retain, use, disclose or otherwise process relevant data for the purpose of performing the service only or as permitted by applicable laws. Before data interaction with third parties, please be informed of their privacy and data security policy.

## Disclaimer

- a) We acknowledge no liability for any injury or damage arising from the reliance upon the information.
- b) We shall bear no liability resulting from any inaccuracies or omissions, or from the use of the information contained herein.
- c) While we have made every effort to ensure that the functions and features under development are free from errors, it is possible that they could contain errors, inaccuracies, and omissions. Unless otherwise provided by valid agreement, we make no warranties of any kind, either implied or express, and exclude all liability for any loss or damage suffered in connection with the use of features and functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage may have been foreseeable.
- d) We are not responsible for the accessibility, safety, accuracy, availability, legality, or completeness of information, advertising, commercial offers, products, services, and materials on third-party websites and third-party resources.

***Copyright © Quectel Wireless Solutions Co., Ltd. 2022. All rights reserved.***

# About the Document

Document Information	
Title	L89 R2.0&LC29H&LC79H Firmware Upgrade Guide
Subtitle	GNSS Module Series
Document Type	Firmware Upgrade Guide
Document Status	Released

## Revision History

Version	Date	Description
-	2021-04-28	Creation of the document
1.0	2021-05-19	First official release
1.1	2021-08-18	Added the applicable module LC79H.
1.2	2022-04-12	<ol style="list-style-type: none"> <li>Added the default UART baud rate for firmware upgrades of the applicable modules.</li> <li>Updated List of Constants and added a note on the format length for LC29H (BA, CA) (Table 1);</li> <li>Added the information about DA files and updated the checksum code of DA file (Chapter 2.1.3);</li> <li>Updated the diagram of Sync with DA and Get DA Report (Figure 7);</li> <li>Updated the diagram of Format Flash (Figure 8);</li> <li>Updated the diagram of Send FW File to Module, added checksum code of the FW packet, and added notes about the FW file and FW packet (Chapter 2.1.7);</li> <li>Updated the upgrade implementation example (Chapter 3);</li> <li>Deleted the chapter of Disconnect Module.</li> </ol>

## Contents

About the Document.....	3
Contents .....	4
Table Index.....	5
Figure Index .....	6
<b>1 Introduction .....</b>	<b>7</b>
<b>2 Firmware Upgrade Process .....</b>	<b>8</b>
2.1. Communication Between Host and Module .....	10
2.1.1. Handshake with Module.....	11
2.1.2. Disable Module WDT .....	12
2.1.3. Send DA File to Module .....	13
2.1.4. Jump to DA.....	14
2.1.5. Sync with DA .....	14
2.1.6. Format Flash .....	16
2.1.7. Send FW File to Module .....	17
<b>3 Upgrade Implementation Example.....</b>	<b>19</b>
<b>4 Appendix Reference .....</b>	<b>27</b>

Table Index

Table 1: List of Constants ..... 8

Table 2: Terms and Abbreviations ..... 27

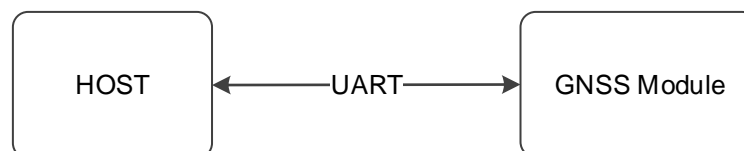
## Figure Index

Figure 1: Firmware Upgrade Connection.....	7
Figure 2: Communication Between Host and Module .....	10
Figure 3: Handshake.....	11
Figure 4: Disable Module WDT .....	12
Figure 5: Send DA File to Module .....	13
Figure 6: Jump to DA .....	14
Figure 7: Sync with DA and Get DA Report.....	15
Figure 8: Format Flash.....	16
Figure 9: Send FW File to Module .....	17

# 1 Introduction

This document introduces the procedure to upgrade the firmware of the Quectel L89 R2.0, LC29H and LC79H GNSS modules.

Following the procedure illustrated in this document, you can upgrade and download the firmware to the target GNSS module via the UART interface. The baud rate for firmware upgrade is 115200 bps by default.



**Figure 1: Firmware Upgrade Connection**



# 2 Firmware Upgrade Process

This chapter describes all the necessary steps in the firmware upgrade process.

The following table defines all the constants used in this document.

**Table 1: List of Constants**

Constant Name	Constant Value	Steps
BROM_ERROR	0x1000	<a href="#">Figure 4: Disable Module WDT</a> <a href="#">Figure 5: Send DA File to Module</a> <a href="#">Figure 6: Jump to DA</a>
Module WDT Register Address	0xA2080000	<a href="#">Figure 4: Disable Module WDT</a>
WDT Value	0x0010	<a href="#">Figure 4: Disable Module WDT</a>
DA Run/Start Address	0x04204000	<a href="#">Figure 5: Send DA File to Module</a> <a href="#">Figure 6: Jump to DA</a>
DA Length	0x00006CF7 (DA file requiring 115200 bps for transmission) 0x00006DCB (DA file requiring 921600 bps for transmission)	<a href="#">Figure 5: Send DA File to Module</a>
Flash Manufacturer ID	0x00EF	<a href="#">Figure 7: Sync with DA and Get DA Report</a>
Flash Device ID1	0x0060	<a href="#">Figure 7: Sync with DA and Get DA Report</a>
Flash Device ID2	0x0016	<a href="#">Figure 7: Sync with DA and Get DA Report</a>
Flash Mount Status	0x00000000	<a href="#">Figure 7: Sync with DA and Get DA Report</a>
Flash Base Address	0x08000000	<a href="#">Figure 7: Sync with DA and Get DA Report</a>

Flash Size	0x00400000	<u>Figure 7: Sync with DA and Get DA Report</u>
Format Physical Address	0x08000000	<u>Figure 8: Format Flash</u>
Format Length <sup>1)</sup>	0x00400000	<u>Figure 8: Format Flash</u>
FW Flash Address	0x08000000 ( <i>partition_table.bin</i> ) 0x08003000 ( <i>ag3335_bootloader.bin</i> ) 0x08013000 ( <i>gnss_demo.bin</i> ) 0x083FF000 ( <i>gnss_config.bin</i> )	<u>Figure 9: Send FW File to Module</u>
FW Packet Length	0x00001000	<u>Figure 9: Send FW File to Module</u>

#### NOTE

1. The bytes to be sent are transmitted in the big-endian format.
2. <sup>1)</sup>When upgrading the firmware of LC29H (BA, CA), the Format Length must be set to 0x003EF000.

## 2.1. Communication Between Host and Module

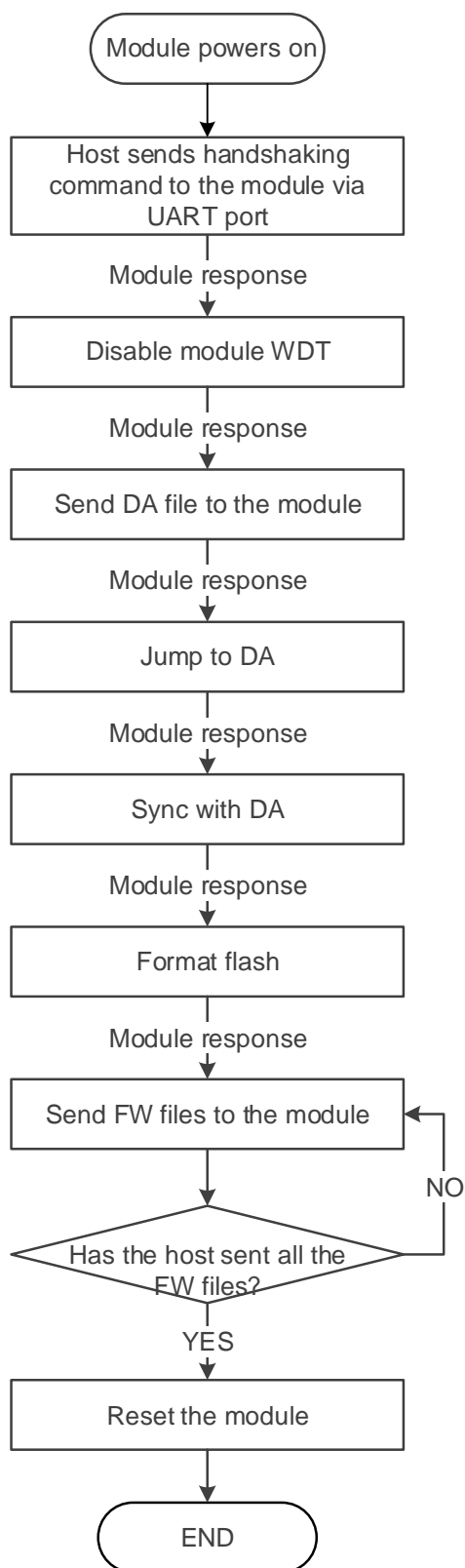


Figure 2: Communication Between Host and Module

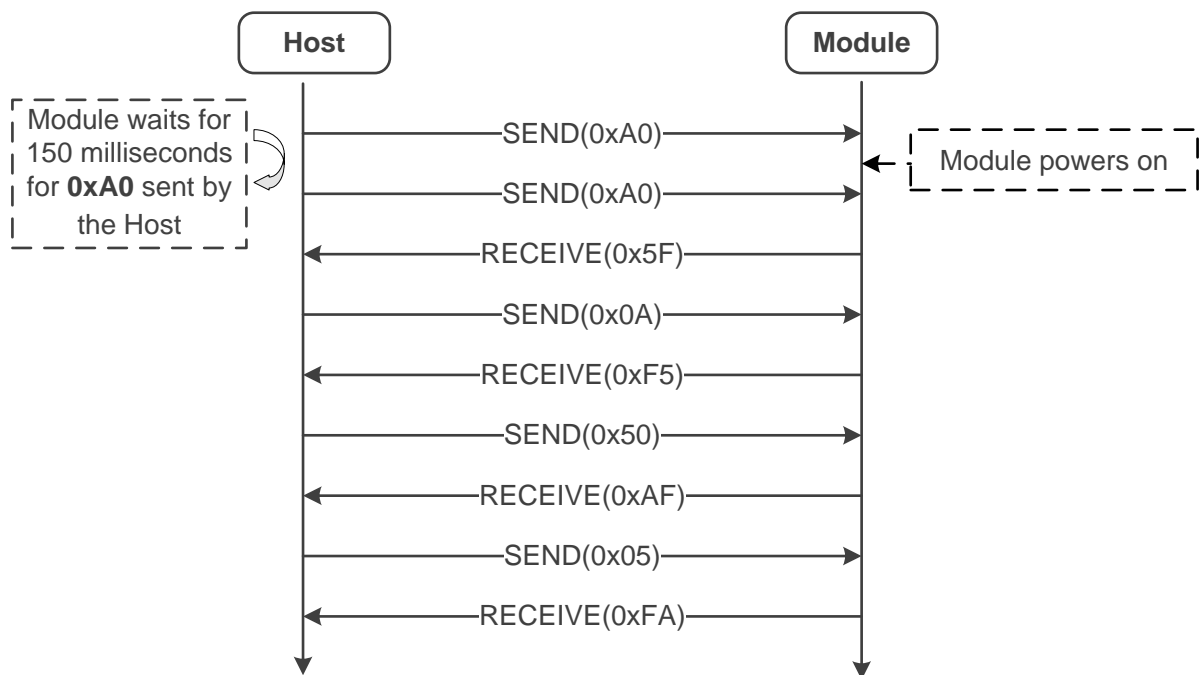
The following sections illustrate how the host establishes communication with the module and sends commands and files to the module:

- Handshake with the module
- Disable module WDT (Watch Dog Timer)
- Send DA (Download Agent) file to the module
- Jump to DA
- Sync with DA
- Format flash
- Send FW (firmware) files to the module

### 2.1.1. Handshake with Module

This section describes the handshaking mechanism for the communication between the host and the module.

After the module is powered on, it waits for 150 milliseconds for **0xA0** sent by the host. If it fails to receive the **0xA0**, the handshake fails.



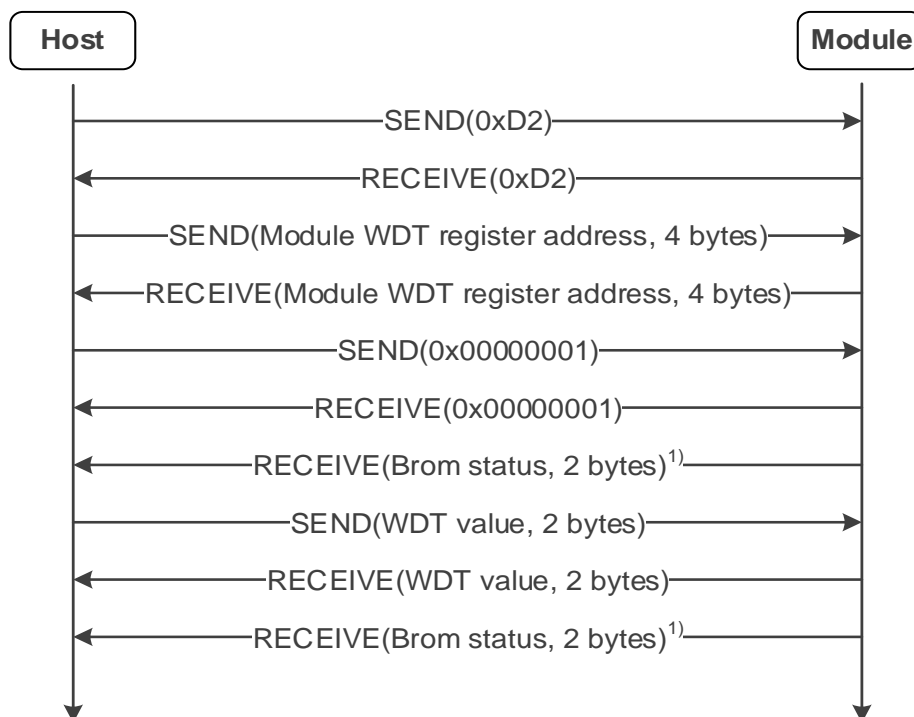
**Figure 3: Handshake**

#### NOTE

The host sends **0xA0** repeatedly until it receives **0x5F** from the module, and then sends **0x0A**.

### 2.1.2. Disable Module WDT

The following diagram illustrates how to disable module WDT to avoid module reboot.



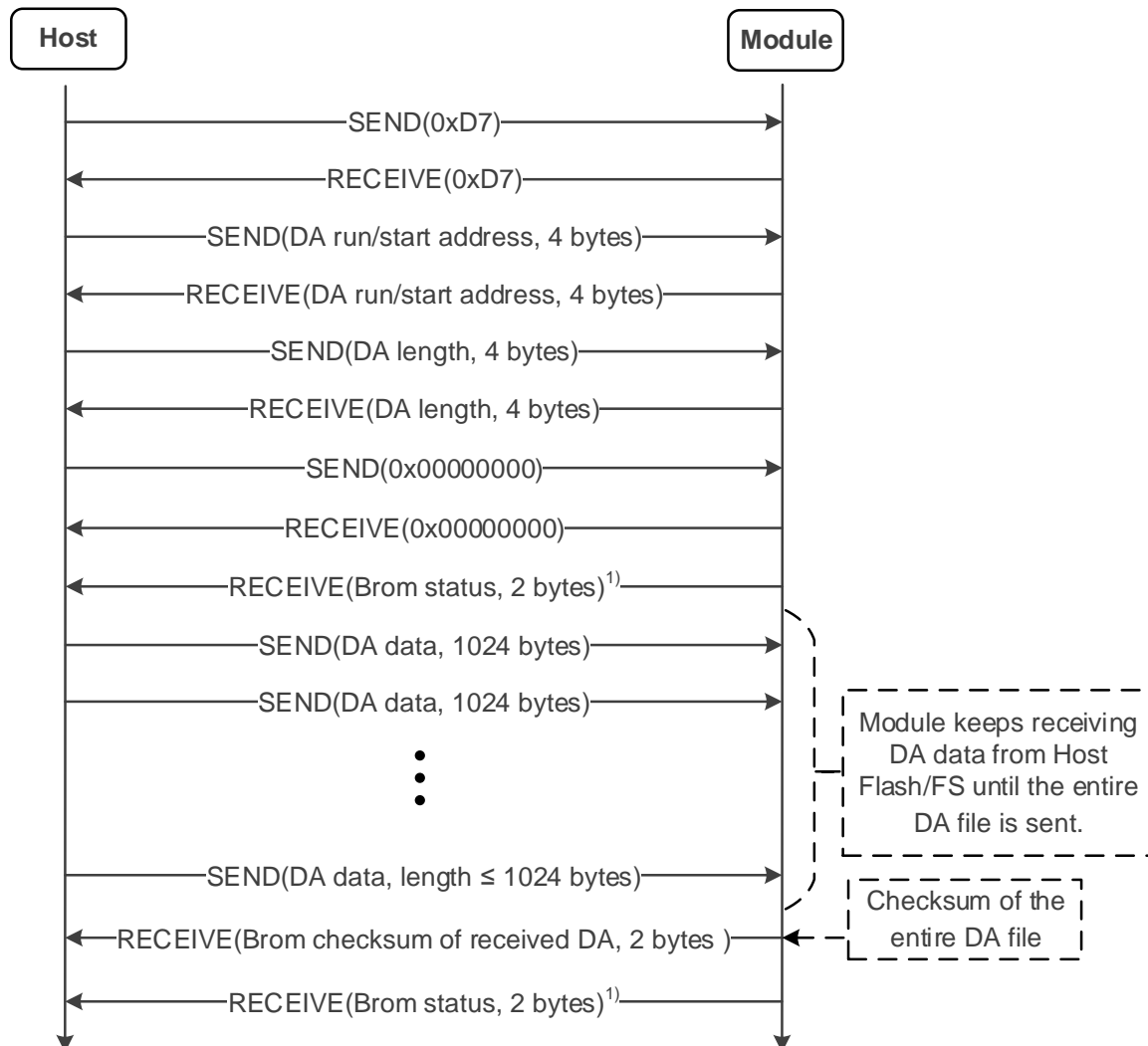
**Figure 4: Disable Module WDT**

#### NOTE

<sup>1)</sup> means Brom status must not be BROM\_ERROR.

### 2.1.3. Send DA File to Module

The following diagram illustrates how to send a DA file to the module. You can use either of the two DA files provided: one requires the baud rate of 921600 bps for transmission, and the other requires the baud rate of 115200 bps.



**Figure 5: Send DA File to Module**

#### NOTE

<sup>1)</sup> means Brom status must not be BROM\_ERROR.

#### Checksum Code of DA File

```

uint16_t DA_compute_checksum (uint8_t *buf, uint32_t buf_len)
{

```

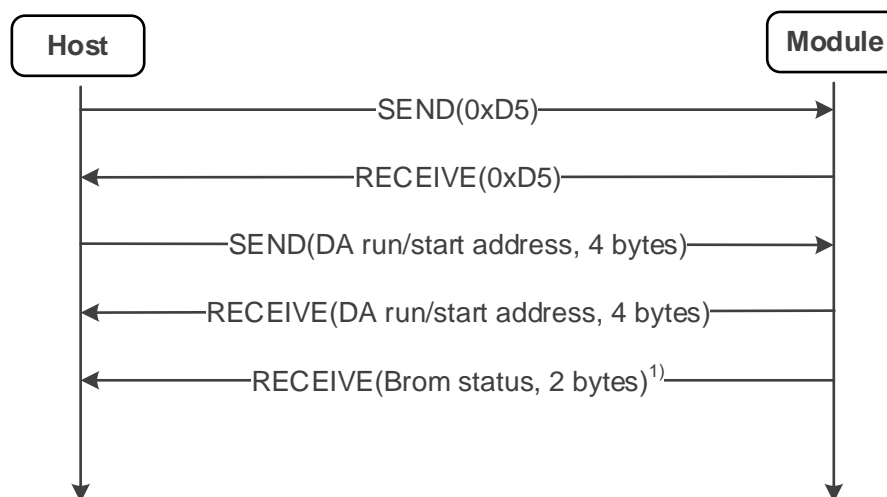
```

uint16_t checksum = 0;
if (buf == NULL || buf_len == 0) {
    return 0;
}
int i = 0;
for (i = 0; i < buf_len / 2; i++) {
    checksum ^= *(uint16_t *) (buf + i * 2);
}
if ((buf_len % 2) == 1) {
    checksum ^= buf[i * 2];
}
return checksum;
}
uint16_t local_check_sum ^= DA_compute_checksum(data_buf, len);

```

#### 2.1.4. Jump to DA

The following diagram illustrates how to inject the DA and how to execute it.



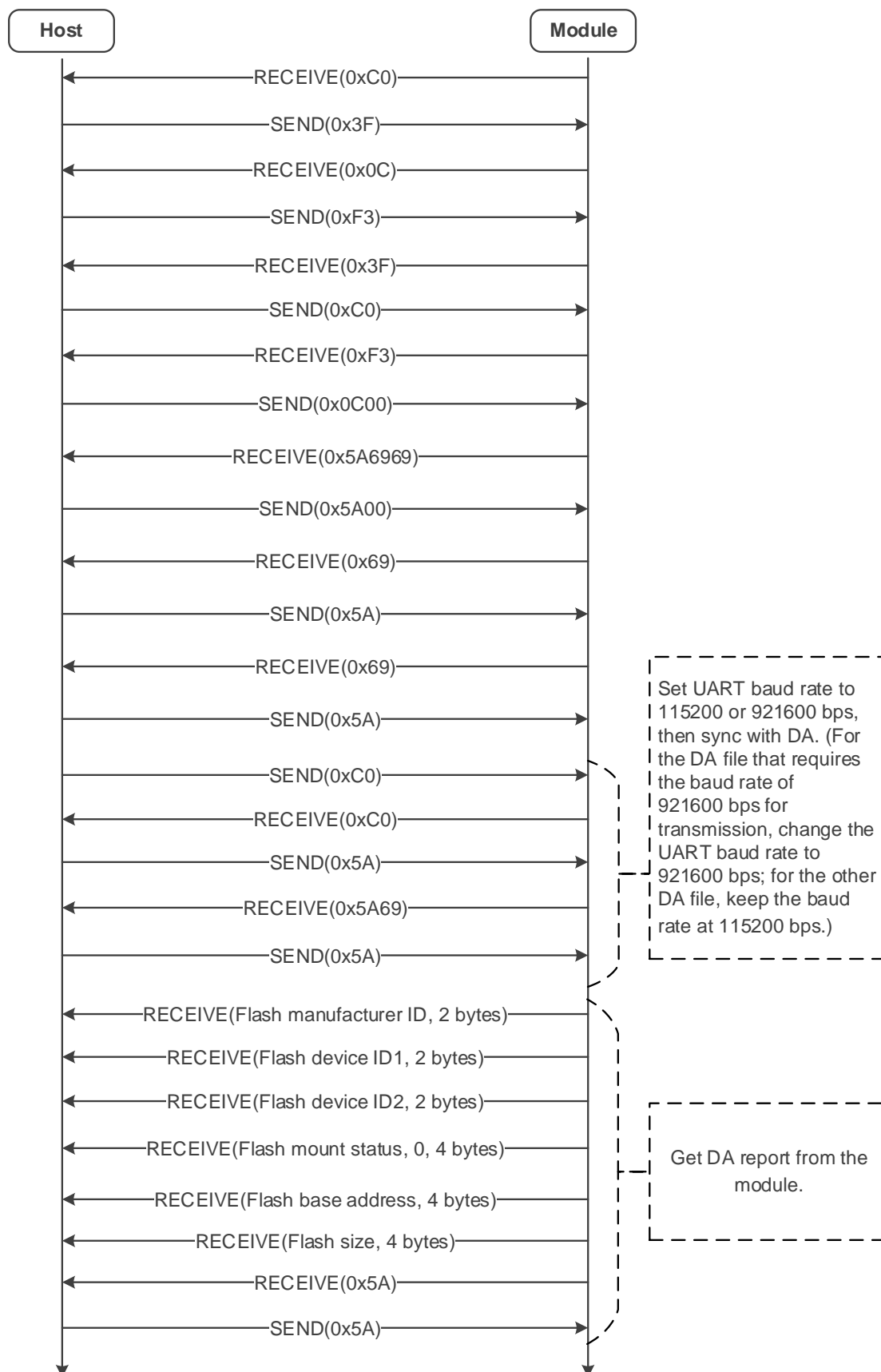
**Figure 6: Jump to DA**

#### NOTE

<sup>1)</sup> means Brom status must not be BROM\_ERROR.

#### 2.1.5. Sync with DA

The following diagram illustrates how to sync with DA and get the DA report in detail.

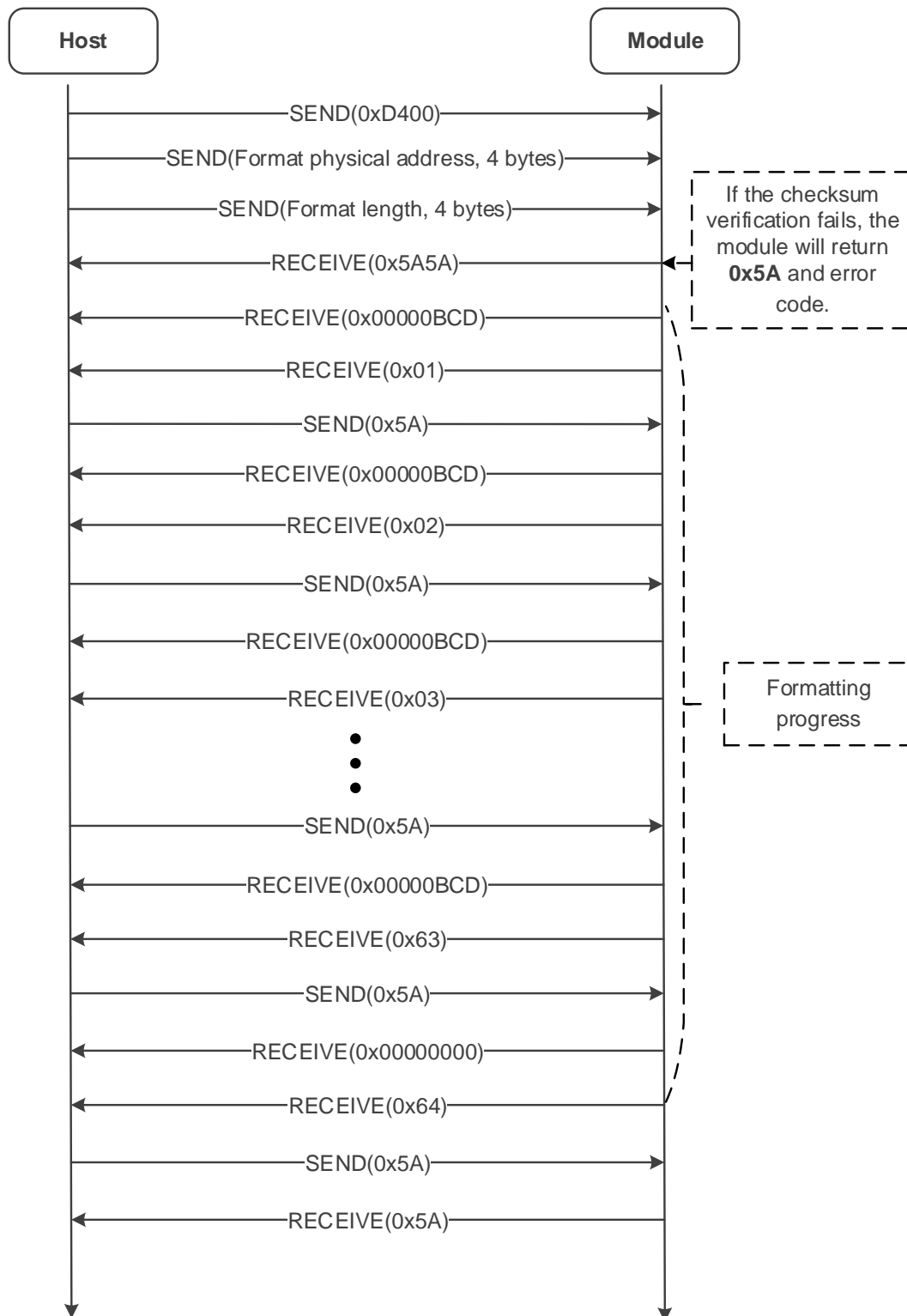


**Figure 7: Sync with DA and Get DA Report**



### 2.1.6. Format Flash

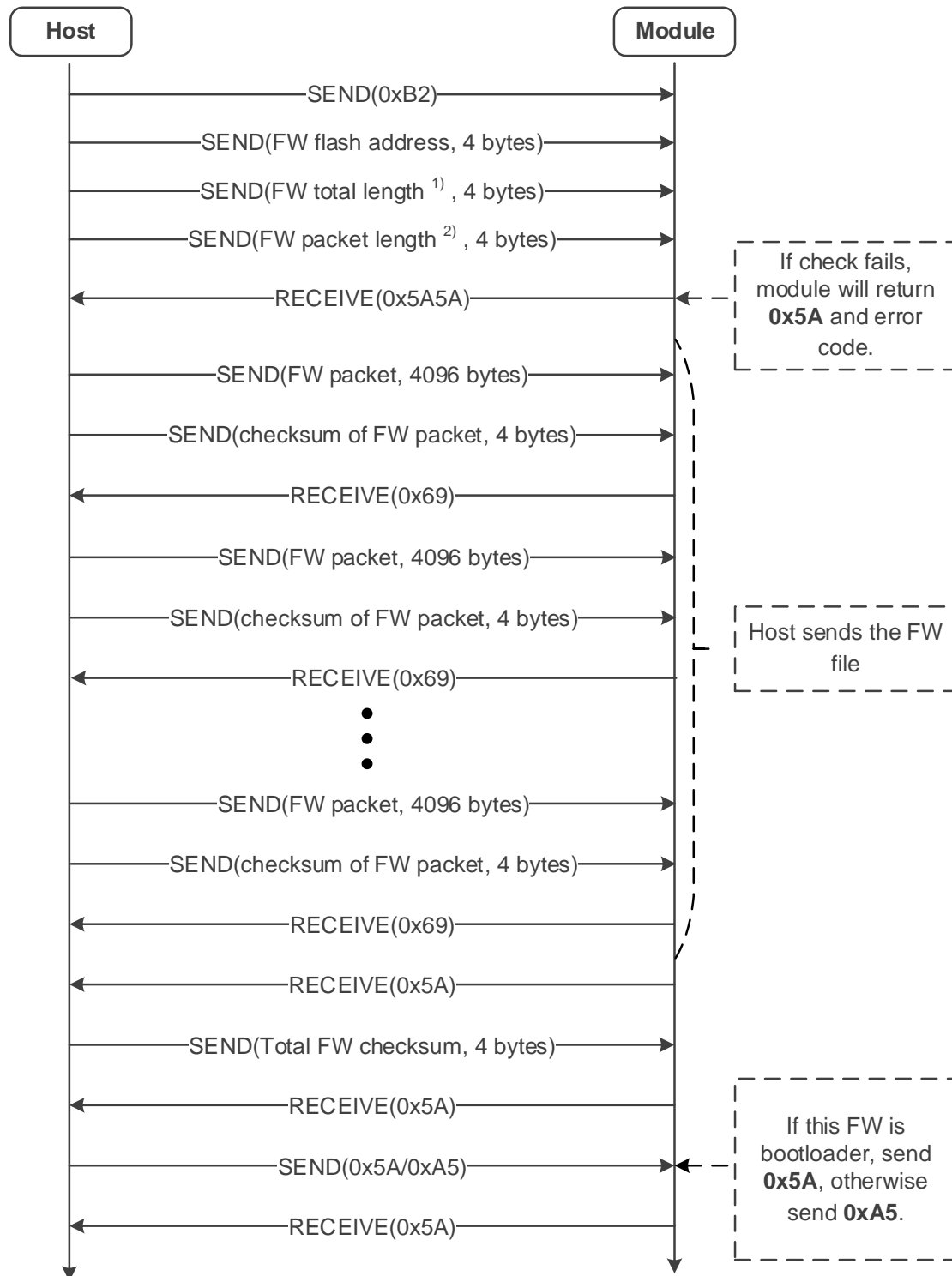
The following diagram illustrates how to format the module's flash.



**Figure 8: Format Flash**

### 2.1.7. Send FW File to Module

The following diagram illustrates how to send FW files to the module.



**Figure 9: Send FW File to Module**

**NOTE**

1. <sup>1)</sup> The FW total length must be an integer multiple of 4K.
2. <sup>2)</sup> Size of each packet is 4096 bytes.

**Checksum Code of the FW Packet**

```
uint32_t CRC_compute_simple_checksum (uint8_t *buf, uint32_t buf_len) {
    uint32_t checksum = 0;
    if (buf == NULL || buf_len == 0) {
        return 0;
    }
    for (int i = 0; i < buf_len; i++) {
        checksum += *(buf + i);
    }
    return checksum;
}

uint32_t packet_checksum = CRC_compute_simple_checksum (buf, buf_len);
//Total Checksum Code of the FW file:
uint32_t Total_FW_checksum += packet_checksum;
```

**NOTE**

If the last FW packet is less than 4096 bytes, add "0xFF" at the end of the last FW packet.

# 3 Upgrade Implementation Example

Below is an example of the L89 R2.0 firmware upgrade.

[//Figure 3: Handshake](#)

//Host keeps sending **0xA0**, and the module is powered on.

A0

A0

A0

...

A0

A0

A0

A0

//Module responds with **0x5F**.

5F

//Host sends **0x0A**.

0A

//Module responds with **0xF5**.

F5

//Host sends **0x50**.

50

//Module responds with **0xAF**.

AF

//Host sends **0x05**.

05

//Module responds with **0xFA**.

FA

[//Figure 4: Disable Module WDT](#)

//Host sends **0xD2**.

D2

//Module responds with **0xD2**.

D2

//Host sends **0xA2080000**.

A2 08 00 00

//Module responds with **0xA2080000**.

A2 08 00 00

//Host sends **0x00000001**.

```

00 00 00 01
//Module responds with 0x00000001.
00 00 00 01
//Module responds with Brom status.
00 01
//Host sends 0x0010.
00 10
//Module responds with 0x0010.
00 10
//Module responds with Brom status.
00 01
//Figure 5: Send DA File to Module
//Host sends 0xD7.
D7
//Module responds with 0xD7.
D7
//Host sends DA run/start address.
04 20 40 00
//Module responds with DA run/start address.
04 20 40 00
//Host sends DA length. (For DA file requiring the baud rate of 115200 bps for transmission, the DA length
is "00 00 6C F7"; for DA file requiring the baud rate of 921600 bps for transmission, the DA length is
"00 00 6D CB".)
00 00 6C F7
//Module responds with DA length.
00 00 6C F7
//Host sends 0x00000000.
00 00 00 00
//Module responds with 0x00000000.
00 00 00 00
//Module responds with Brom status.
00 00
//Host sends DA data, in a chunk of 1 KB.
27 B6 11 49 11 4A 4F F0 00 00 91 42 00 F0 05 80 41 F8 04 0B 91 42 7F F6 .....
//Host sends DA data, in a chunk of 1 KB.
23 48 08 3E 1F 46 89 46 92 46 FF F7 95 FF 10 23 33 60 34 68 1F 4D E3 06 .....
.....
//Host sends the last chunk of DA data. (length ≤ 1 KB)
65 73 73 28 30 78 25 78 29 20 74 6F 20 62 6C 6F 63 6B 20 69 6E 64 65 78 .....
//Module responds with Brom checksum of the received DA (2 bytes).
46 1D (Checksum Code of DA File)
//Module responds with Brom status.
00 00
//Figure 6: Jump to DA

```

```
//Host sends 0xD5.
D5
//Module responds with 0xD5.
D5
//Host sends DA run/start address.
04 20 40 00
//Module responds with DA run/start address.
04 20 40 00
//Module responds with Brom status.
00 00
//Figure 7: Sync with DA
//Module responds with 0xC0.
C0
//Host sends 0x3F.
3F
//Module responds with 0x0C.
0C
//Host sends 0xF3.
F3
//Module responds with 0x3F.
3F
//Host sends 0xC0.
C0
//Module responds with 0xF3.
F3
//Host sends 0x0C00.
0C 00
//Module responds with 0x5A6969.
5A 69 69
//Host sends 0x5A00.
5A 00
//Module responds with 0x69.
69
//Host sends 0x5A.
5A
//Module responds with 0x69.
69
//Host sends 0x5A.
5A
//Host sends 0xC0. (For the DA file that requires the baud rate of 921600 bps for transmission, change
the UART baud rate to 921600 bps; for the other DA file, keep the baud rate at 115200 bps.)
C0
//Module responds with 0xC0.
C0
```

```

//Host sends 0x5A.
5A
//Module responds with 0x5A69.
5A 69
//Host sends 0x5A.
5A
//Module responds with flash manufacturer ID.
00 EF
//Module responds with flash device ID1.
00 60
//Module responds with flash device ID2.
00 16
//Module responds with flash mount status.
00 00 00 00
//Module responds with flash base address.
08 00 00 00
//Module responds with flash size.
00 40 00 00
//Module responds with 0x5A.
5A
//Host sends 0x5A.
5A
// Figure 8: Format Flash
//Host sends 0xD400.
D4 00
//Host sends format physical address.
08 00 00 00
//Host sends format length. (When upgrading the firmware of LC29H (BA, CA), the format length must be
set to "00 3E F0 00".)
00 40 00 00
//Module responds with 0x5A5A.
5A 5A
//Module responds with 0x00000BCD and formatting progress.
00 00 0B CD 01
//Host sends 0x5A and waits for the module to respond with 0x00000BCD and formatting progress.
5A 00 00 0B CD 02
//Host sends 0x5A and waits for the module to respond with 0x00000BCD and formatting progress.
5A 00 00 0B CD 03
//Host sends 0x5A and waits for the module to respond with 0x00000BCD and formatting progress.
5A 00 00 0B CD 04
.....
//Host sends 0x5A and waits for the module to respond with 0x00000BCD and formatting progress.
5A 00 00 0B CD 63
//Host sends 0x5A and waits for the module to respond with 0x00000000 and formatting progress.

```

```

5A 00 00 00 00 64
//Host sends 0x5A.
5A
//Module responds with 0x5A.
5A
//Figure 9: Send FW File to Module
//Host sends 0xB2.
B2
//Host sends FW flash address (partition_table.bin).
08 00 00 00
//Host sends FW total length (partition_table.bin).
00 00 10 00
//Host sends FW packet length (4 bytes).
00 00 10 00
//Module responds with 0x5A5A.
5A 5A
//Host sends FW packet (4096 bytes).
00 00 00 00 00 00 00 00 00 00 00 00 00 00 10 00 08 00 00 00 00 00 10 00 00 .....
//Host sends checksum of FW packet (4 bytes).
00 0E 7B 22 (Checksum Code of the FW Packet)
//Module responds with 0x695A.
69 5A
//Host sends the total FW checksum (4 bytes).
00 0E 7B 22 (Checksum Code of the FW Packet)
//Module responds with 0x5A.
5A
//Host sends 0x5A/0xA5. (If this FW is ag3335_bootloader.bin, send 0x5A, otherwise send 0xA5.)
A5
//Module responds with 0x5A.
5A
//Host sends 0xB2.
B2
//Host sends FW flash address (ag3335_bootloader.bin).
08 00 30 00
//Host sends FW total length (ag3335_bootloader.bin). (4K alignment: the FW total length must be an
integer multiple of 4K.)
00 00 50 00
//Host sends FW packet length (4 bytes).
00 00 10 00
//Module responds with 0x5A5A.
5A 5A
//Host sends FW packet (4096 bytes).
DF F8 3C D0 72 B6 00 20 03 21 0E 4A 42 F8 04 0B 11 60 0D 4A 0D 4B 00 F0 .....
//Host sends checksum of FW packet (4 bytes).

```



```

00 05 AD 0C (Checksum Code of the FW Packet)
//Module responds with 0x69.
69
//Host sends FW packet (4096 bytes).
D4 30 00 20 10 BD 00 BF C7 28 00 04 4F F4 40 73 A1 F8 BC 30 4F F4 80 73 .....
//Host sends checksum of FW packet (4 bytes).
00 05 67 E0 (Checksum Code of the FW Packet)
//Module responds with 0x69.
69
//Host sends FW packet (4096 bytes).
65 6C 20 70 61 73 73 21 0D 0A 00 72 69 73 69 6E 67 20 66 72 65 71 20 74 .....
//Host sends checksum of FW packet (4 bytes).
00 05 D5 53 (Checksum Code of the FW Packet)
//Module responds with 0x69.
69
//Host sends FW packet (4096 bytes).
70 B5 D0 F8 C8 30 DA 06 04 46 0D 46 01 D4 FF F7 54 FF D4 F8 C8 30 DB 06 .....
//Host sends checksum of FW packet (4 bytes).
00 06 D9 83 (Checksum Code of the FW Packet)
//Module responds with 0x69.
69
//Host sends FW packet (4096 bytes).
06 48 04 E0 06 48 02 E0 06 48 00 E0 06 48 00 EB 81 00 FF F7 B3 BF 6F F0 .....
//Host sends checksum of FW packet (4 bytes).
00 0D 0D 06 (Checksum Code of the FW Packet)
//Module responds with 0x695A.
69 5A
//Host sends the total FW checksum (4 bytes).
00 24 D0 C8 (Checksum Code of the FW Packet)
//Module responds with 0x5A.
5A
//Host sends 0x5A/0xA5. (If this FW is ag3335_bootloader.bin, send 0x5A, otherwise send 0xA5.)
5A
//Module responds with 0x5A.
5A
//Host sends 0xB2.
B2
//Host sends FW flash address (gnss_demo.bin).
08 01 30 00
//Host sends FW total length (gnss_demo.bin). (4K alignment: the FW total length must be an integer
multiple of 4K.)
00 11 50 00
//Host sends FW packet length (4 bytes).
00 00 10 00

```

```
//Module responds with 0x5A5A.  
5A 5A  
//Host sends FW packet (4096 bytes).  
DF F8 A0 D0 72 B6 28 48 80 47 28 48 28 49 01 60 28 48 29 49 01 60 29 48 .....  
//Host sends checksum of FW packet (4 bytes).  
00 06 A7 58 (Checksum Code of the FW Packet)  
//Module responds with 0x69.  
69  
//Host sends FW packet (4096 bytes).  
C3 80 60 46 89 1A B1 FB F6 F2 AD B2 06 FB 12 11 45 EA 01 45 07 FB 02 F7 .....  
//Host sends checksum of FW packet (4 bytes).  
00 06 26 F9 (Checksum Code of the FW Packet)  
//Module responds with 0x69.  
69  
//Host sends FW packet (4096 bytes).  
20 46 0A 46 0B 46 00 F0 EF F9 3D E0 01 21 20 46 00 F0 C2 FA 24 49 2A 46 .....  
//Host sends checksum of FW packet (4 bytes).  
00 06 25 D6 (Checksum Code of the FW Packet)  
//Module responds with 0x69.  
69  
.....  
//Host sends FW packet (4096 bytes).  
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....  
//Host sends checksum of FW packet (4 bytes).  
00 0F F0 00 (Checksum Code of the FW Packet)  
//Module responds with 0x695A.  
69 5A  
//Host sends the total FW checksum (4 bytes).  
07 B8 3C 46 (Checksum Code of the FW Packet)  
//Module responds with 0x5A.  
5A  
//Host sends 0x5A/0xA5. (If this FW is ag3335_bootloader.bin, send 0x5A, otherwise send 0xA5.)  
A5  
//Module responds with 0x5A.  
5A  
//Host sends 0xB2.  
B2  
//Host sends FW flash address (gnss_config.bin).  
08 3F F0 00  
//Host sends FW total length (gnss_config.bin). (4K alignment: the FW total length must be an integer multiple of 4K.)  
00 00 10 00  
//Host sends FW packet length (4 bytes).  
00 00 10 00
```

```
//Module responds with 0x5A5A.
5A 5A
//Host sends FW packet (4096 bytes).
01 01 01 FF FF FF FF FF FF FF FF FF FF FF FF FF FF 01 64 00 00 00 FF FF FF .....
//Host sends checksum of FW packet (4 bytes).
00 0E 46 9E (Checksum Code of the FW Packet)
//Module responds with 0x695A.
69 5A
//Host sends the total FW checksum (4 bytes).
00 0E 46 9E (Checksum Code of the FW Packet)
//Module responds with 0x5A.
5A
//Host sends 0x5A/0xA5. (If this FW is ag3335_bootloader.bin, send 0x5A, otherwise send 0xA5.)
A5
//Module responds with 0x5A.
5A
```

# 4 Appendix Reference

**Table 2: Terms and Abbreviations**

Abbreviation	Description
CRC	Cyclic Redundancy Check
DA	Download Agent
FS	File System
FW	Firmware
GNSS	Global Navigation Satellite System
UART	Universal Asynchronous Receiver/Transmitter
WDT	Watchdog Timer