LC76F

Firmware Upgrade Guide

GNSS Module Series

Version: 1.0.

Date: 2021-09-14

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

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.

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.

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.

About the Document

Document Information	
Title	**LC76F Firmware Upgrade Guide**
Subtitle	GNSS Module Series
Document Type	Firmware Upgrade Guide
Document Status	Released

Revision History

Version	Date	Description
-	2021-06-25	Creation of the document
1.0	2021-09-14	First official release

Contents

Table Index

Figure Index

1 Introduction

This document explains how to upgrade the firmware on Quectel LC76F GNSS module.

Following the procedure illustrated in this document, you can upgrade and download the firmware to the target GNSS module via the UART interface.

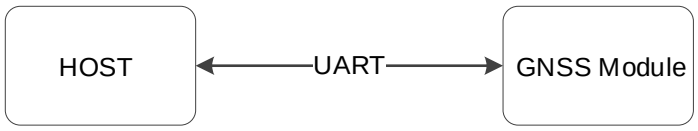

Figure 1: Firmware Upgrade Connection

2 Firmware Upgrade Process

This chapter explains 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
NMEA_START_CMD	$PGKC75*1D\r\n	*Figure 3: Handshake*
BOOT_ROM_WRITE_CMD	0xA1	*Figure 4: Send BE*
BOOT_ROM_CHECKSUM_CMD	0xA4	*Figure 4: Send BE*
BE_START_ADDR	0x00000C00	*Figure 4: Send BE* *Figure 5: Jump to BE*
BOOT_ROM_JUMP_CMD	0xA8	*Figure 5: Jump to BE*
SYNC_CHAR	0xB0	*Figure 6: Sync with BE*
BE_MAJOR_VER	0x01	*Figure 6: Sync with BE*
BE_MINOR_VER	0x00	*Figure 6: Sync with BE*
BE_SPEED_CMD	0xE1	*Figure 6: Sync with BE*
BAUDRATE_FULL_SYNC_COUNT	0X00	*Figure 6: Sync with BE*
ACK	0x6B	*Figure 6: Sync with BE* *Figure 7: Download Firmware*
FW_START_ADDR	0x00000000	*Figure 7: Download Firmware*
FW_END_ADDR	0x0008E00B	*Figure 7: Download Firmware*
BE_MEM_CMD	0xE2	*Figure 7: Download Firmware*
BE_WRITE_CMD	0xE3	*Figure 7: Download Firmware*
BE_FINISH_CMD	0xE5	*Figure 7: Download Firmware*

PACKET_LENGTH	0x00000400	*Figure 7: Download Firmware*
CONT_CHAR	0xB9	*Figure 7: Download Firmware*
NMEA_CLEAR_CMD	$PGKC41*1A\r\n	*Figure 8: Reset*
NMEA_RESET_CMD	$PGKC30,3,1*1E\r\n	*Figure 8: Reset*

Table 2: Baudrate

Constant Name	Baudrate ID	Baudrate (bps)
UART_BAUD_921600	0x01	921600
UART_BAUD_460800	0x02	460800
UART_BAUD_230400	0x03	230400
UART_BAUD_115200	0x04	115200
UART_BAUD_57600	0x05	57600
UART_BAUD_38400	0x06	38400
UART_BAUD_19200	0x07	19200
UART_BAUD_14400	0x08	14400
UART_BAUD_9600	0x09	9600

NOTE

The bytes to be sent are transmitted in the big-endian format.

2.1. Communication Between Host and Module

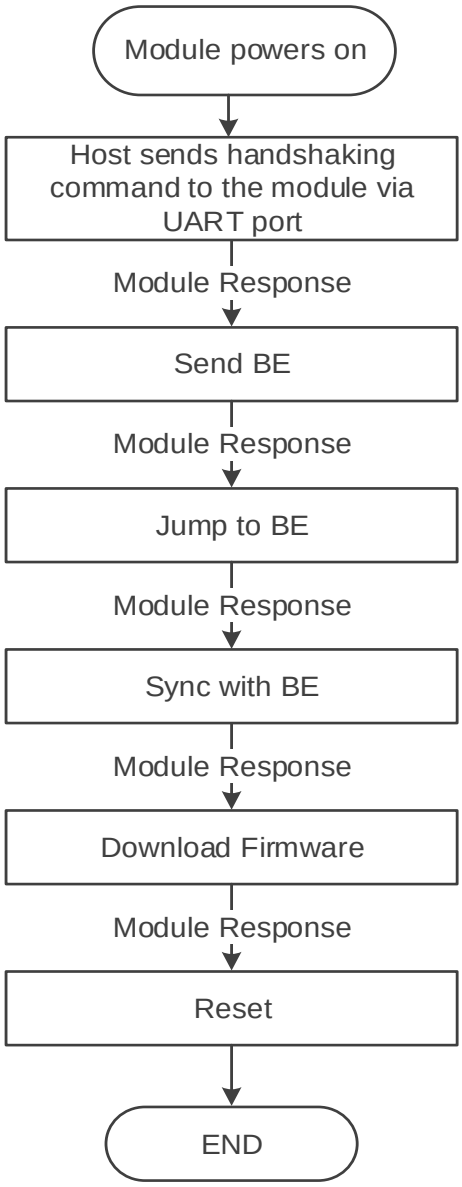

Figure 2: Communication Between Host and Module

The following sections provide a guide on how the host communicates with module and sends commands and files to the module:

- Handshake with module
- Send BE (Burn Engine)
- Jump to BE
- Sync with BE
- Download firmware
- Reset

2.1.1. Handshake with Module

This section describes the handshake protocol in detail.

Host can send command **$PGKC75*1D** make the module reset. After the module is reset, it will wait for "0x90" sent by the host for 300 milliseconds. If the "0x90" is not received, the handshake fails.

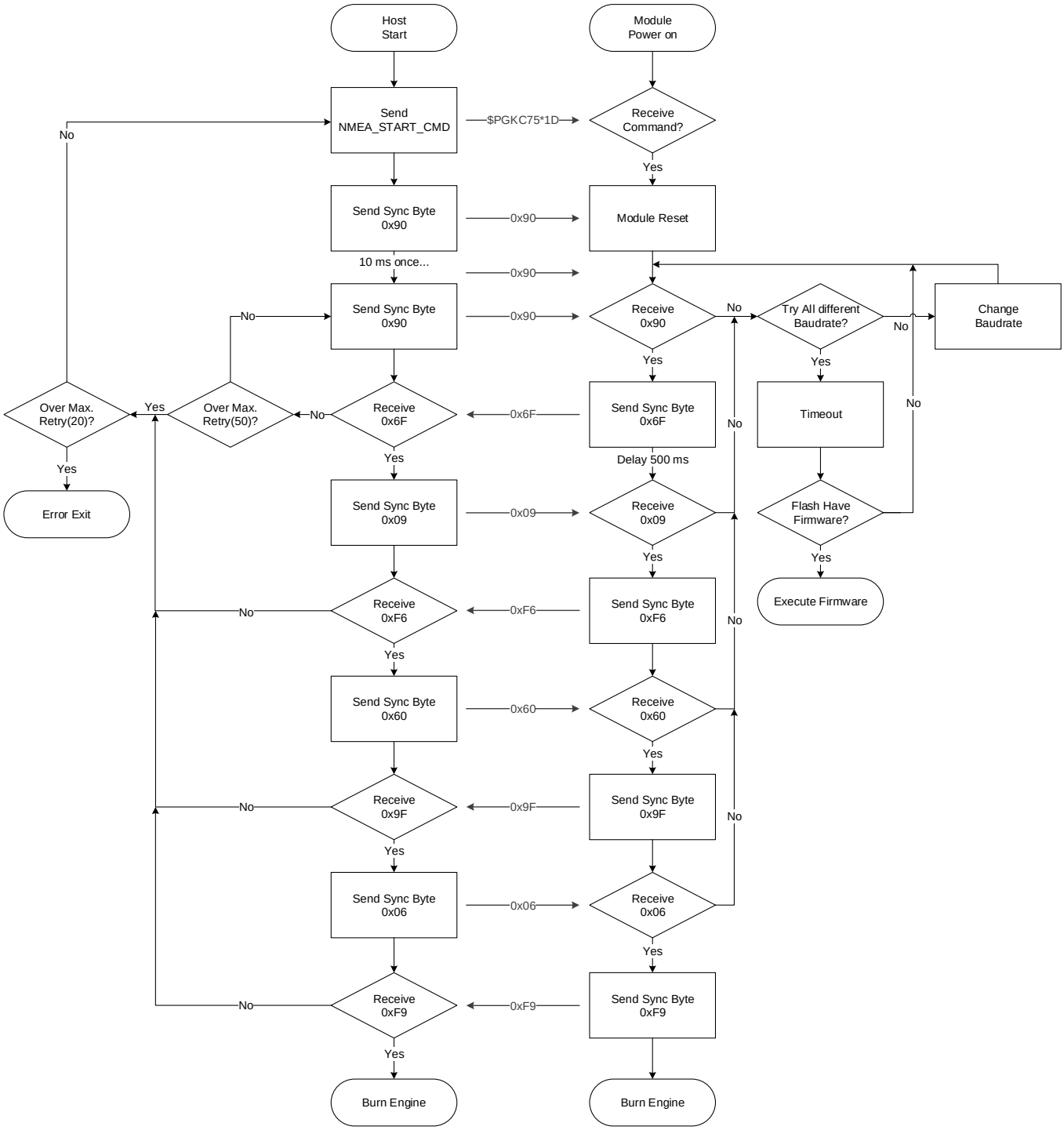

Figure 3: Handshake

2.1.2. Send BE

This section describes how to send the BE to the module.

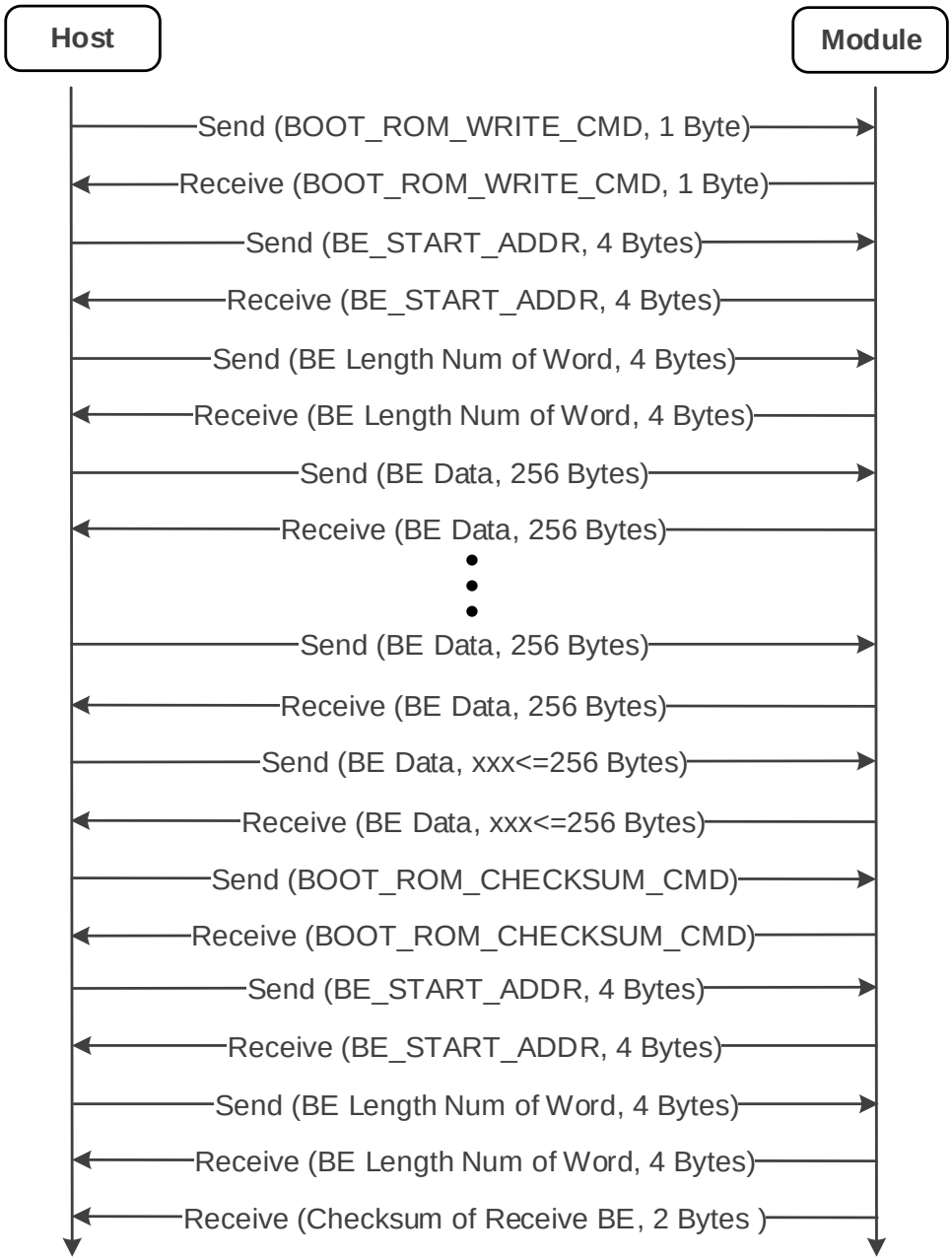

Figure 4: Send BE

2.1.3. Jump to BE

This section describes how to jump to BE and run it.

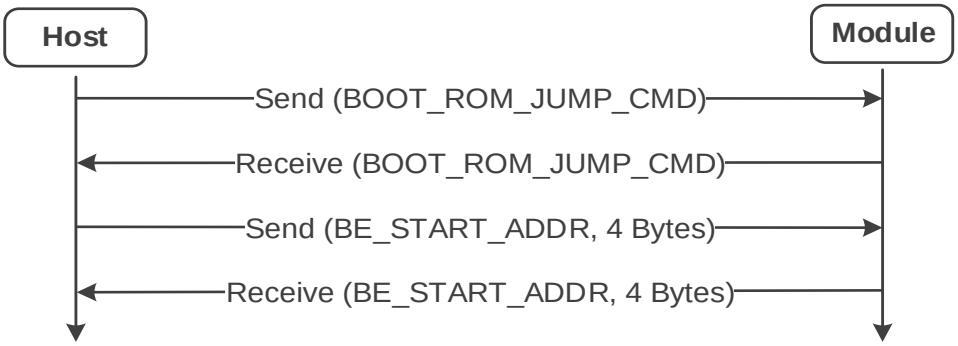

Figure 5: Jump to BE

2.1.4. Sync with BE

This section describes how to sync with BE in detail.

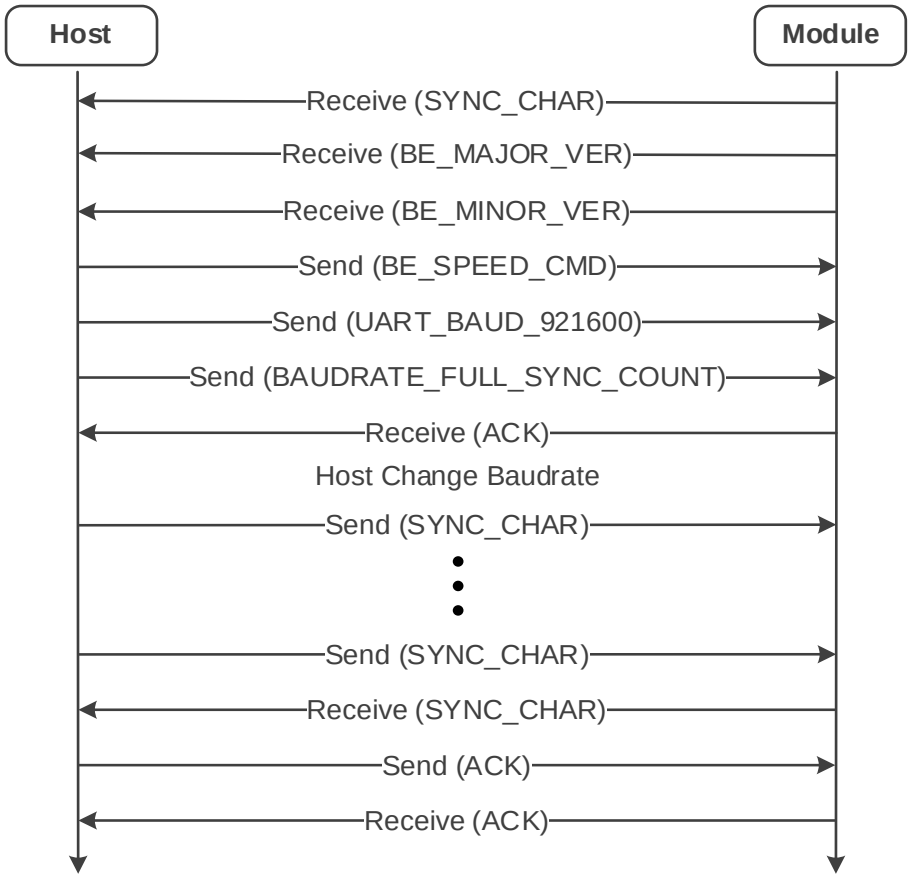

Figure 6: Sync with BE

2.1.5. Download Firmware

This section describes how to download firmware to flash.

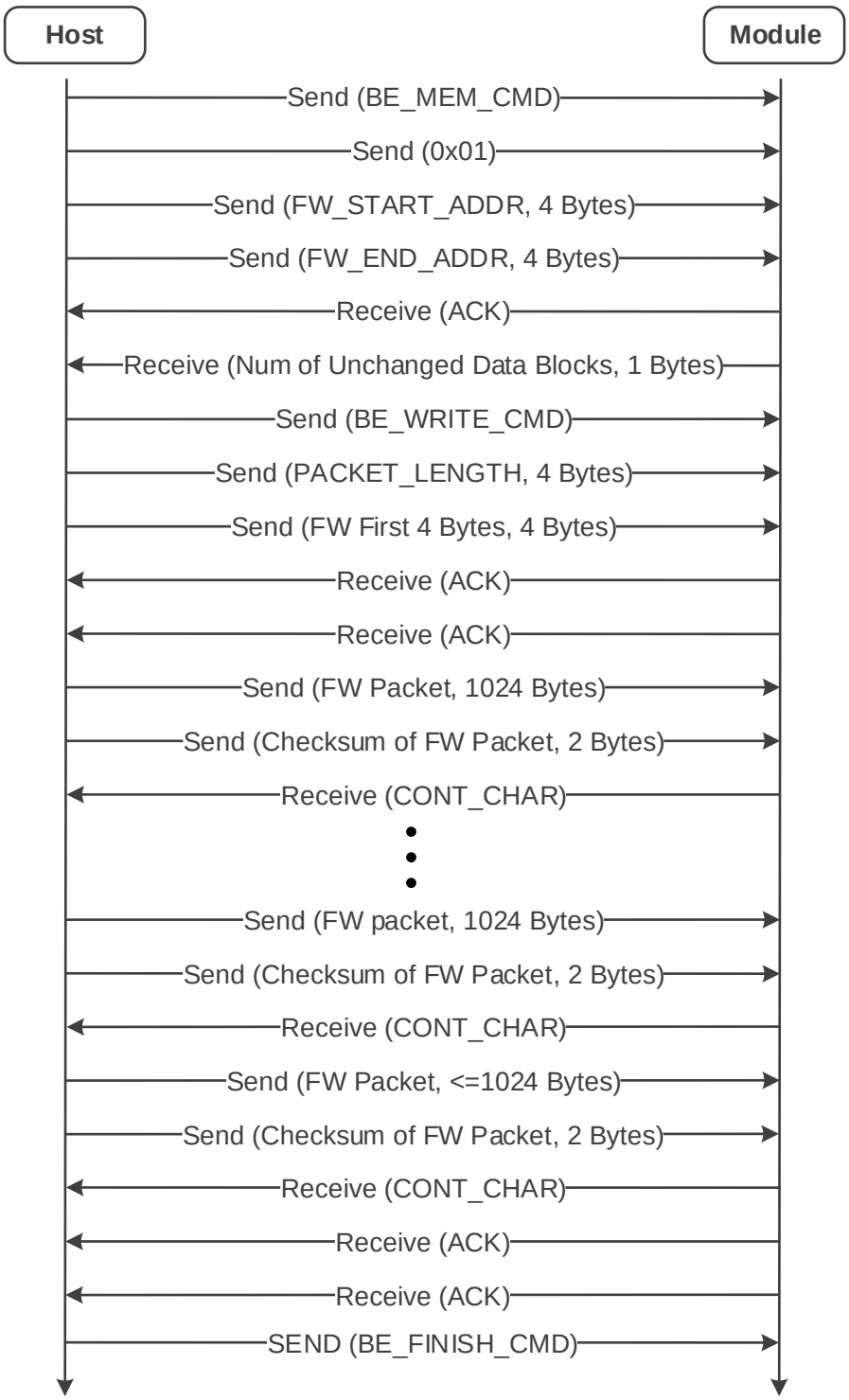

Figure 7: Download Firmware

2.1.6. Reset

This section describes how to reset the module.

After firmware download process is finished, the module will output NMEA message normally. Host will try to receive NMEA message in different baud rate. Once host receive NMEA head "$", it will send **NMEA_CLEAR_CMD** and **NMEA_RESET_CMD** to reset the module in current baud rate. The firmware upgrade is completed after the module is reset.

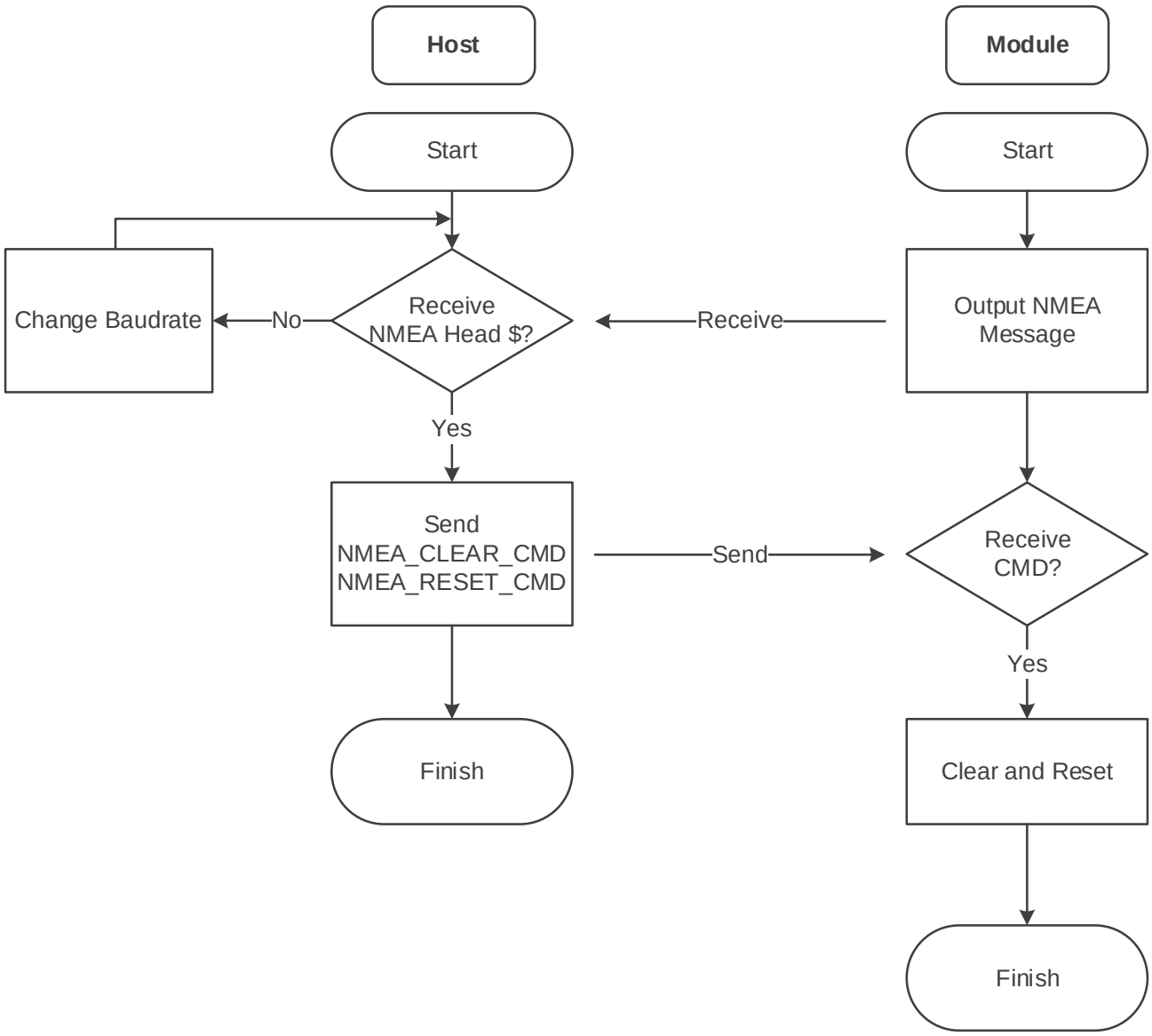

Figure 8: Reset

2.2. Sample Code for Checksum

Checksum Code of BE File

```
uint16_t BE_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;
}
```

Checksum Code of FW Packet

```
uint32_t FW_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;
}
```

3 Upgrade Implementation Example

In this chapter you can find the example of firmware upgrade procedure.

```
//Host sends NMEA_START_CMD
//$PGKC75*1D
24 50 47 4B 43 37 35 2A 31 44 0D 0A
//Module will reset later
//Host continuously sends 0x90
90
//Module outputs NMEA message as before
45
//Host continuously sends 0x90
90
//Module outputs NMEA message as before
0D
//Host continuously sends 0x90
90
//Module outputs NMEA message as before
0D
//Host continuously sends 0x90
90
//Module resets
//Host continuously sends 0x90
90
90
//Module responds with 0x6F
6F
//Host sends 0x09
09
//Module responds with 0x6F as before
6F
//Module receives 0x09 and responds with 0xF6
F6
//Host sends 0x60
60
//Module responds with 0x9F
9F
```

//Host sends **0x06**
06
//Module responds with **0xF9**
F9
//Host sends **BOOT_ROM_WRITE_CMD 0xA1**
A1
//Module responds with **0xA1**
A1
//Host sends **BE_START_ADDR**
00 00 0C 00
//Module responds with **BE_START_ADDR**
00 00 0C 00
//Host sends BE Length Num of Word
00 00 11 80
//Module responds with BE Length Num of Word
00 00 11 80
//Host sends first Burn Engine packet (256 Bytes)
00 00 E1 A0 00 00 E1 0F 10 C0 E3 A0 00 01 E1 80 …
//Module responds with first Burn Engine packet (256 Bytes)
00 00 E1 A0 00 00 E1 0F 10 C0 E3 A0 00 01 E1 80 …
//Host sends second Burn Engine packet (256 Bytes)
20 00 E0 82 00 00 20 83 00 04 21 50 30 08 85 91 …
//Module responds with second Burn Engine packet (256 Bytes)
20 00 E0 82 00 00 20 83 00 04 21 50 30 08 85 91 …
…
//Host sends last Burn Engine packet (<= 256 Bytes)
00 00 00 40 00 00 00 00 00 08 00 00 20 00 00 00 …
//Module responds with last Burn Engine packet (<=256 Bytes)
00 00 00 40 00 00 00 00 00 08 00 00 20 00 00 00 …
//Host sends **BOOT_ROM_CHECKSUM_CMD**
A4
//Module responds with **BOOT_ROM_CHECKSUM_CMD**
A4
//Host sends **BE_START_ADDR**
00 00 0C 00
//Module responds with **BE_START_ADDR**
00 00 0C 00
//Host sends BE Length Num of Word
00 00 11 80
//Module responds with BE Length Num of Word
00 00 11 80
//Module responds with checksum
40 69
//Host sends **BOOT_ROM_JUMP_CMD**

A8
//Module responds with **BOOT_ROM_JUMP_CMD**
A8
//Host sends **BE_START_ADDR**
00 00 0C 00
//Module responds with **BE_START_ADDR**
00 00 0C 00
//Module responds with **SYNC_CHAR**
B0
//Module responds with **BE_MAJOR_VER**
01
//Module responds with **BE_MINOR_VER**
00
//Host sends **BE_SPEED_CMD**
E1
//Host sends **UART_BAUD_921600**
01
//Host sends **BAUDRATE_FULL_SYNC_COUNT**
00
//Module responds with **ACK**
6B
//Host changes baud rate to 921600
//Host sends **SYNC_CHAR**
B0
//Module tries to respond with **SYNC_CHAR** in difference baud rate
00
//Host sends **SYNC_CHAR**
B0
//Module tries to respond with **SYNC_CHAR** in difference baud rate
00
//Host sends **SYNC_CHAR**
B0
//Module tries to respond with **SYNC_CHAR** in difference baud rate
AA
//Host sends **SYNC_CHAR**
B0
//Module tries to respond with **SYNC_CHAR** in difference baud rate
//Sync success
B0
//Host sends **ACK**
6B
//Module responds with **ACK**
6B
//Host sends **BE_MEM_CMD**

E2
//Host sends **0x01**
01
//Host sends **FW_START_ADDR**
00 00 00 00
//Host sends **FW_END_ADDR**
00 08 E0 0B
//Module responds with **ACK**
6B
//Module responds with number of unchanged data blocks
00
//Host sends **BE_WRITE_CMD**
E3
//Host sends packet length
//0x00000400 = 1024 Bytes
00 00 04 00
//Host sends firmware first 4 bytes
82 DA 39 D6
//Module responds with **ACK** when burn engine has saved unchanged data
6B
//Module responds with **ACK** when sector erase done
6B
//Host sends first firmware packet except first 4 Bytes (1024 Bytes)
F9 CD F9 B7 DE 21 2E 3C 94 58 17 58 95 71 2D 83 …
//Host sends checksum of first firmware packet (2 Bytes)
F2 4D
//Module responds with **CONT_CHAR** (continue char)
B9
//Host sends second firmware packet (1024 Bytes)
C2 B7 FA 50 50 75 7B 3C 49 45 96 BF 7E E9 97 52 …
//Host sends checksum of first firmware packet (2 Bytes)
F4 27
//Module responds with **CONT_CHAR** (continue char)
B9
…
//Host sends last firmware packet (<= 1024 Bytes)
7B AD BA 9B 90 94 8D 57
//Host sends checksum of last firmware packet (2 Bytes)
04 85
//Module responds with **CONT_CHAR** (continue char)
B9
//Module responds with **ACK** when firmware upgrade was done
6B
//Module responds with **ACK** when checksum was done

6B
//Host sends **BE_FINISH_CMD**
E5
//Host changes baud rate to 9600
//Module sends NMEA messages (**$GNTXT,GK9501*7C\r\n**)
24 47 4E 54 58 54 2C 47 4B 39 35 30 31 2A 37 43 0D 0A
//Host sends **NMEA_CLEAR_CMD** (**$PGKC41*1A\r\n**)
24 50 47 4B 43 34 31 2A 31 41 0D 0A
//Host sends **NMEA_RESET_CMD** (**$PGKC30,3,1*1E\r\n**)
24 50 47 4B 43 33 30 2C 33 2C 31 2A 31 45 0D 0A

4 Appendix Reference

Table 3: Terms and Abbreviations

Abbreviation	Description
ACK	Acknowledge
BE	Burn Engine
CMD	Command
FW	Firmware
GNSS	Global Navigation Satellite System
NMEA	NMEA (National Marine Electronics Association) 0183 Interface Standard
UART	Universal Asynchronous Receiver/Transmitter