Telit Jupiter

MT33xx Host Flash Application Note

80433NT11333A r.2 – 2015-04-02

Making machines talk.

APPLICABILITY TABLE

PRODUCT
SL871
SL869-V2
SL868-A
SC872-A

SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE

Notice

While reasonable efforts have been made to assure the accuracy of this document, Telit assumes no liability resulting from any inaccuracies or omissions in this document, or from use of the information obtained herein. The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies or omissions. Telit reserves the right to make changes to any products described herein and reserves the right to revise this document and to make changes from time to time in content hereof with no obligation to notify any person of revisions or changes. Telit does not assume any liability arising out of the application or use of any product, software, or circuit described herein; neither does it convey license under its patent rights or the rights of others.

It is possible that this publication may contain references to, or information about Telit products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that Telit intends to announce such Telit products, programming, or services in your country.

Copyrights

This instruction manual and the Telit products described in this instruction manual may be, include or describe copyrighted Telit material, such as computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and its licensors certain exclusive rights for copyrighted material, including the exclusive right to copy, reproduce in any form, distribute and make derivative works of the copyrighted material. Accordingly, any copyrighted material of Telit and its licensors contained herein or in the Telit products described in this instruction manual may not be copied, reproduced, distributed, merged or modified in any manner without the express written permission of Telit. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppels, or otherwise, any license under the copyrights, patents or patent applications of Telit, as arises by operation of law in the sale of a product.

Computer Software Copyrights

The Telit and 3rd Party supplied Software (SW) products described in this instruction manual may include copyrighted Telit and other 3rd Party supplied computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and other 3rd Party supplied SW certain exclusive rights for copyrighted computer programs, including the exclusive right to copy or reproduce in any form the copyrighted computer program. Accordingly, any copyrighted Telit or other 3rd Party supplied SW computer programs contained in the Telit products described in this instruction manual may not be copied (reverse engineered) or reproduced in any manner without the express written permission of Telit or the 3rd Party SW supplier. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppels, or otherwise, any license under the copyrights, patents or patent applications of Telit or other 3rd Party supplied SW, except for the normal non-exclusive, royalty free license to use that arises by operation of law in the sale of a product.

Usage and Disclosure Restrictions

License Agreements

The software described in this document is the property of Telit and its licensors. It is furnished by express license agreement only and may be used only in accordance with the terms of such an agreement.

Copyrighted Materials

Software and documentation are copyrighted materials. Making unauthorized copies is prohibited by law. No part of the software or documentation may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, without prior written permission of Telit

High Risk Materials

Components, units, or third-party products used in the product described herein are NOT fault-tolerant and are NOT designed, manufactured, or intended for use as on-line control equipment in the following hazardous environments requiring fail-safe controls: the operation of Nuclear Facilities, Aircraft Navigation or Aircraft Communication Systems, Air Traffic Control, Life Support, or Weapons Systems (High Risk Activities"). Telit and its supplier(s) specifically disclaim any expressed or implied warranty of fitness for such High Risk Activities.

Trademarks

TELIT and the Stylized T Logo are registered in Trademark Office. All other product or service names are the property of their respective owners.

Table of Contents

Table of Figures

1. Introduction

1.1. Scope

This document describes the GNSS feature known as Extended Ephemeris (EE), with particular focus on the application support of Telit's MT3333 based GNSS modules with the context of Server-Generated Extended Ephemeris (SGEE) feature.

- SL871
- SL869-V2
- SL868-A
- SC872-A

To be specific, the SGEE feature that is found in the above family modules is supplied by Mediatek® in their Extended Prediction Orbit (EPO) data. With reference to SGEE feature in those modules, Telit SGEE-EPO is hereinafter referring to the Mediatek's EPO.

This document is intended for software engineers and developers who are interested in implementing the SGEE feature within their devices.

It is also useful for those who are interested in learning more about the Assisted GNSS capability that is found in the Telit's GNSS module families that support the features.

1.2. Contact Information, Support

For general contact, technical support, to report documentation errors and to order manuals, contact Telit Technical Support Center (TTSC) at:

TS-EMEA@telit.com

TS-AMERICAS@telit.com

TS-LATINAMERICA@telit.com

TS-APAC@telit.com

Alternatively, use:

http://www.telit.com/en/products/technical-support-center/contact.php

For detailed information about where you can buy the Telit modules or for recommendations on accessories and components visit:

http://www.telit.com

To register for product news and announcements or for product questions contact Telit Technical Support Center (TTSC).

Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.

Telit appreciates feedback from the users of our information.

1.3. Text Conventions

STOP

<u>Danger – This information MUST be followed or catastrophic equipment failure or bodily injury may occur.</u>

⚠

Caution or Warning – Alerts the user to important points about integrating the module, if these points are not followed, the module and end user equipment may fail or malfunction.

ℹ

Tip or Information – Provides advice and suggestions that may be useful when integrating the module.

All dates are in ISO 8601 format, i.e. YYYY-MM-DD.

1.4. Related Documents

SL871 and SL869-V2 Families.

2. Overview

This document provides Application Note on how to write software module to implement a firmware update procedure for Telit's GNSS modules that are based on Mediatek's Brom protocols.

The applicable GNSS modules include:

- SL871
- SL869-V2
- SL868-A
- SC872-A

Telit and GNSS device vendors collaborate to furnish to customers who use Telit's GNSS modules to achieve the goal of improving TTFF performance.

3. Firmware Update Procedure

3.1. Host Device and BootROM Hanshake Protocol

This is a sequence of handshaking /bytes-exchange between the host device and the GNSS module.

3.2. Start Procedure

3.2.1. Command: CMD_Start

This is the first command sent to GNSS module to start the update procedure.

In this command the host will sequentially set its baudrate for serial communication and send the NMEA_START_CMD ($PMTK180) at each baudrate to force the target reset. Then the host will settle at the highest or desirable baudrate (i.e. 115200) that is also supported by the GNSS module to start the update process.

The sequence is started by a number of predefined values in byte-exchange manner with the GNSS module, to ensure that the communication is established and the update mode is set at the GNSS module.

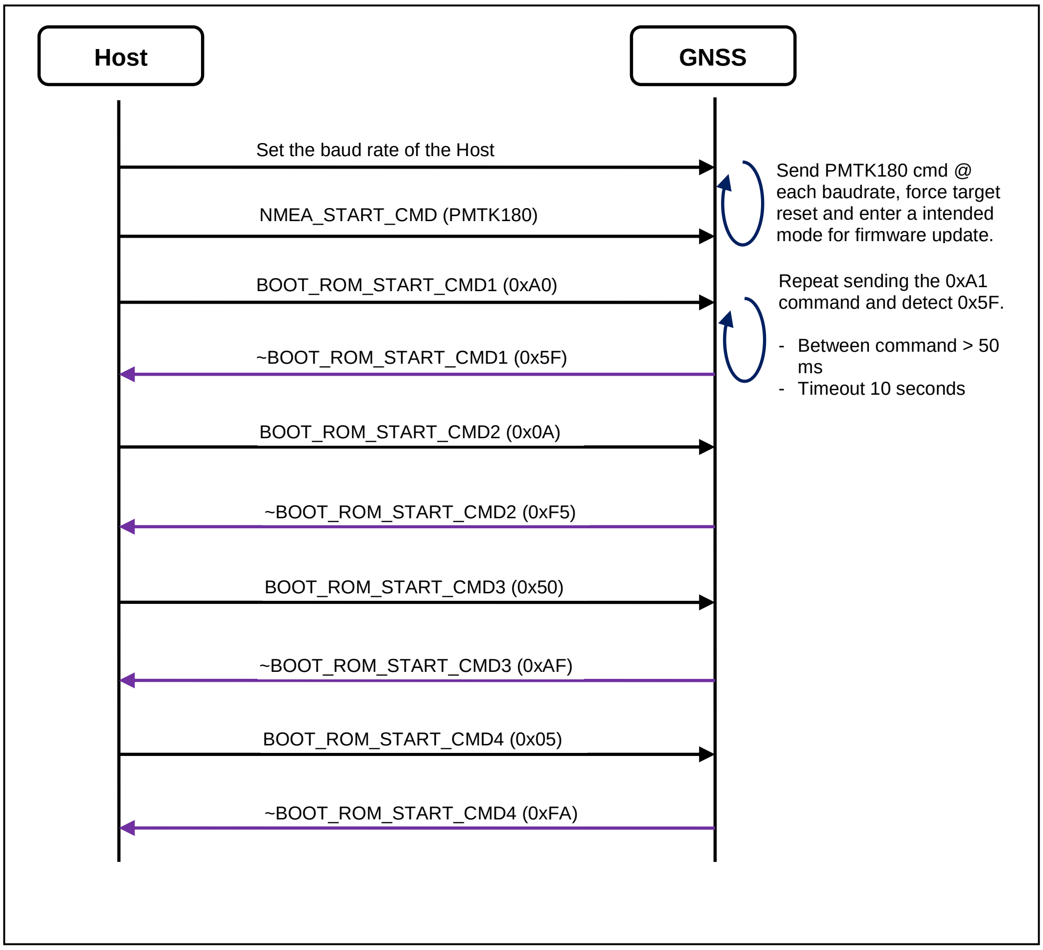
Host
GNSS
Set the baud rate of the Host
NMEA_START_CMD (PMTK180)
Send PMTK180 cmd @ each baudrate, force target reset and enter a intended mode for firmware update.
BOOT_ROM_START_CMD1 (0xA0)
~BOOT_ROM_START_CMD1 (0x5F)
Repeat sending the 0xA1 command and detect 0x5F.
- Between command > 50 ms
- Timeout 10 seconds
BOOT_ROM_START_CMD2 (0x0A)
~BOOT_ROM_START_CMD2 (0xF5)
BOOT_ROM_START_CMD3 (0x50)
~BOOT_ROM_START_CMD3 (0xAF)
BOOT_ROM_START_CMD4 (0x05)
~BOOT_ROM_START_CMD4 (0xFA)

Figure 1: Start procedure - flow of CMD_Start

- Set the baud rate:
- Setting the baudrate is to set the baudrate at the host, to enable sending the NMEA_START_CMD to the GNSS module.
- Among the baud rates for serial port communication (nominal from 4800 and up), the baudrate 115200 is the recommended value.
- BOOT_ROM_START_CMD1 (0xA0):
- This command needs to be sent repeatedly, if the expected ~BOOT_ROM_START_CMD1 (0x5F) is not received by the host.
- The nominal interval between each resend of the command is 50ms.

3.2.2. Command: CMD_Write

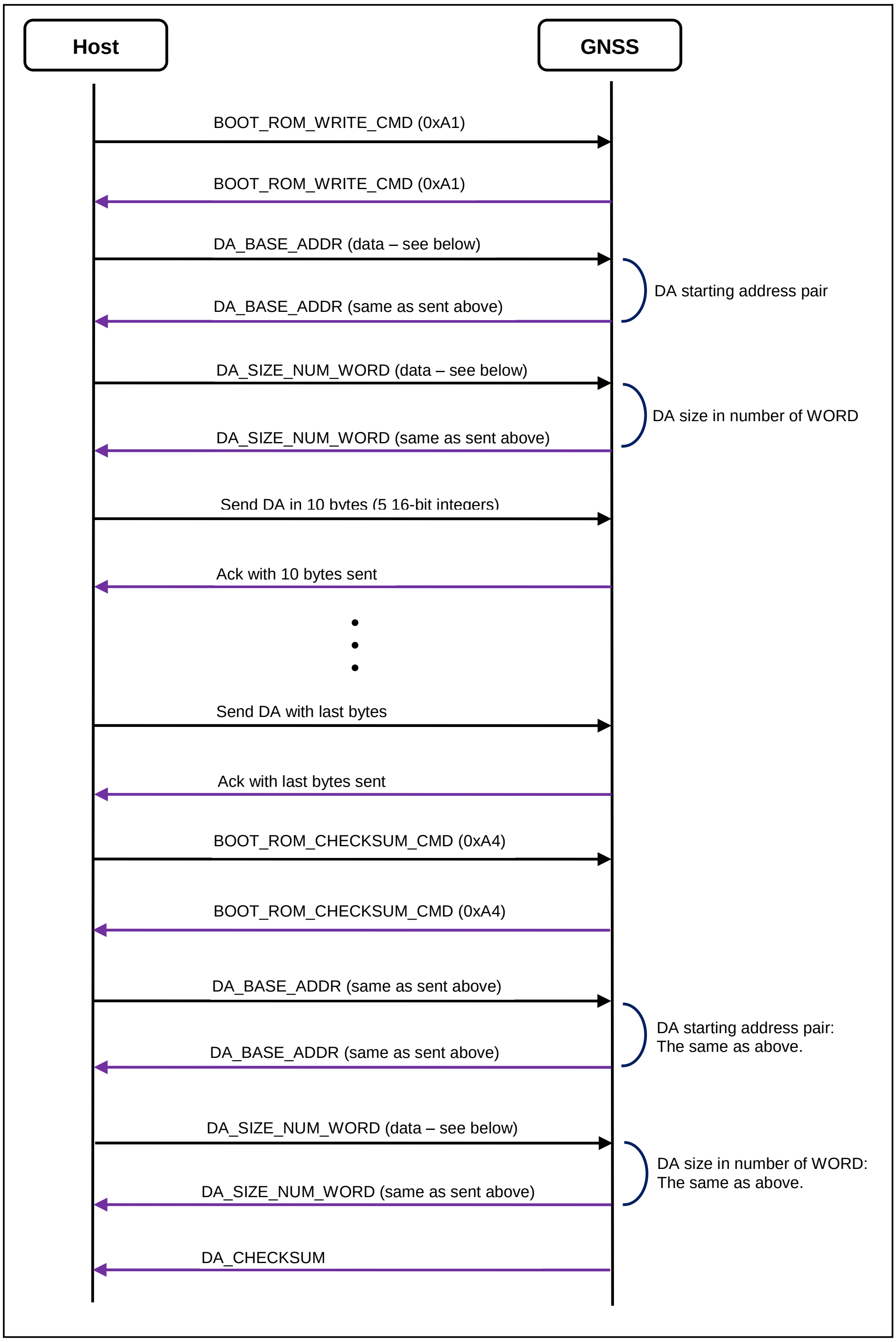
Host
GNSS
BOOT_ROM_WRITE_CMD (0xA1)
BOOT_ROM_WRITE_CMD (0xA1)
DA_BASE_ADDR (data – see below)
DA_BASE_ADDR (same as sent above)
DA starting address pair
DA_SIZE_NUM_WORD (data – see below)
DA_SIZE_NUM_WORD (same as sent above)
DA size in number of WORD
Send DA in 10 bytes (5 16-bit integers)
Ack with 10 bytes sent
Send DA with last bytes
Ack with last bytes sent
BOOT_ROM_CHECKSUM_CMD (0xA4)
BOOT_ROM_CHECKSUM_CMD (0xA4)
DA_BASE_ADDR (same as sent above)
DA_BASE_ADDR (same as sent above)
DA starting address pair: The same as above.
DA_SIZE_NUM_WORD (data – see below)
DA_SIZE_NUM_WORD (same as sent above)
DA size in number of WORD: The same as above.
DA_CHECKSUM

Figure 2: Start procedure - flow of CMD_Write

Data arguments:

- DA_BASE_ADDR

For modules that is MT3329 or later, MT3333 included, the DA starting address is 3072 (0x0000 0C00):

DA_BASE_ADDR (0x00 0x00 0x0C 0x00)

- DA_SIZE_NUM_WORD

The DA size in WORDs = 5946 (0x0000 173A) (bytes = 23784)

DA_SIZE_NUM_WORD (0x00 0x00 0x17 0x3A)

- DA data format

- The DA data is sent in packets of 5 16-bit unsigned integers. That means if it happens that the size of the DA data in bytes is not an even number – there will be a a solo byte to be sent at the end of the DA file - the downloading code has to form a 2-byte value with padding a 0x00 value as the high byte, to satisfy the protocol requirement.

- It is not required, however, the data must be sent in 5 such integers as specified above. The protocol allows that the remaining integers (the number of integers can be < 5) are sent at the end of the DA file.

- DA_CHECKSUM

- A simple exclusive-OR algorithm is applied all bytes of a DA data file that are sent to the module to get the 16-bit checksum values generated. Based on this requirement, pertaining the flow chart above, this checksum does not apply to the DA_BASE_ADDR command and the DA_SIZE_NUM_WORD command.

3.3. Download Download Agent (DA)

3.3.1. Command: CMD_Jump

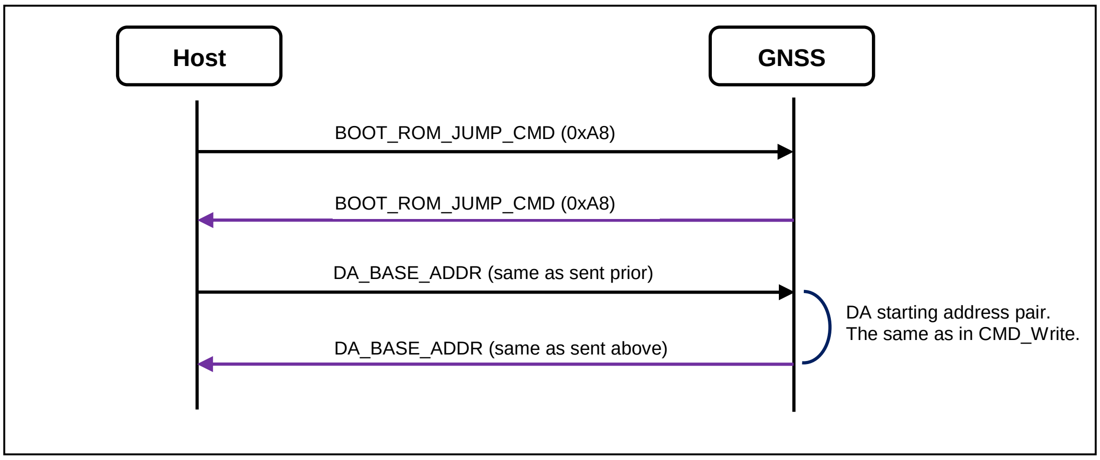
Host
GNSS
BOOT_ROM_JUMP_CMD (0xA8)
BOOT_ROM_JUMP_CMD (0xA8)
DA_BASE_ADDR (same as sent prior)
DA_BASE_ADDR (same as sent above)
DA starting address pair.
The same as in CMD_Write.

Figure 3: Download DA - flow CMD_Jump

3.3.2. DA Information Report

When DA was downloaded and executed, it will first report 20 bytes data with the following information:

(1) SYNC_CHAR (one byte)

(2) DA_VERSION (two bytes)

(3) FLASH_DEVICE_ID (one byte)

(4) FLASH_SIZE (four bytes)

(5) FLASH_HW_ID (eight bytes)

(6) EXT_SRAM_SIZE (four bytes).

- SYNC_CHAR:

When DA was executed, it will return SYNC_CHAR (0xC0). If the return byte isn’t SYNC_CHAR, which means it’s possibly downloading a wrong DA.

- DA_VERSION:

After SYNC_CHAR, DA will return DA’s version number to BROM DLL, it contains two bytes, one is major version, and the other is minor version. BROM DLL will check whether if it supports this DA.

- FLASH_DEVICE_ID:

After report the DA version, DA will automatically detect the flash type on target. If DA supports this flash, then the flash device id will return to BROM DLL. If DA doesn’t support this flash, it will return 0xFF to indicate unknown flash type.

- FLASH_SIZE:

Four bytes flash size, for example: 128Mbits (16Mbytes) flash will be 0x01000000 bytes; DA will send 0x01, 0x00, 0x00, and 0x00.

- FLASH_MANUFACTURE_CODE:

Two bytes flash manufacture code. (Users should refer to the datasheet for each flash.)

- FLASH_DEVICE_CODE:

Two bytes flash device code. (Users should refer to the datasheet for each flash.)

- FLASH_EXT_DEVICE_CODE1:

Two bytes flash extended device code1. (Users should refer to the datasheet for each flash.)

- FLASH_EXT_DEVICE_CODE2:

Two bytes flash extended device code2. (Users should refer to the datasheet for each flash.)

- EXT_SRAM_SIZE:

Four bytes external SRAM size, for example: 64Mbits (8Mbytes) external SRAM will be 0x00800000 bytes; DA will send 0x00, 0x80, 0x00, and 0x00.

3.3.3. Command: CMD_SetMemBlock

This command is used to notify DA the total memory block count and the range for each block. The memory block information indicates how many BIN files will be downloaded and the range of each BIN file.

If any memory block exceed the flash size, DA will return NACK (0xA5) immediately to indicate the DA_MEM_CMD command is fail.

If all the download memory blocks are valid, DA will return ACK (0x5A) and UNCHANGED_BLOCK_COUNT to notify PC side how many unchanged block should be recovery after downloading.

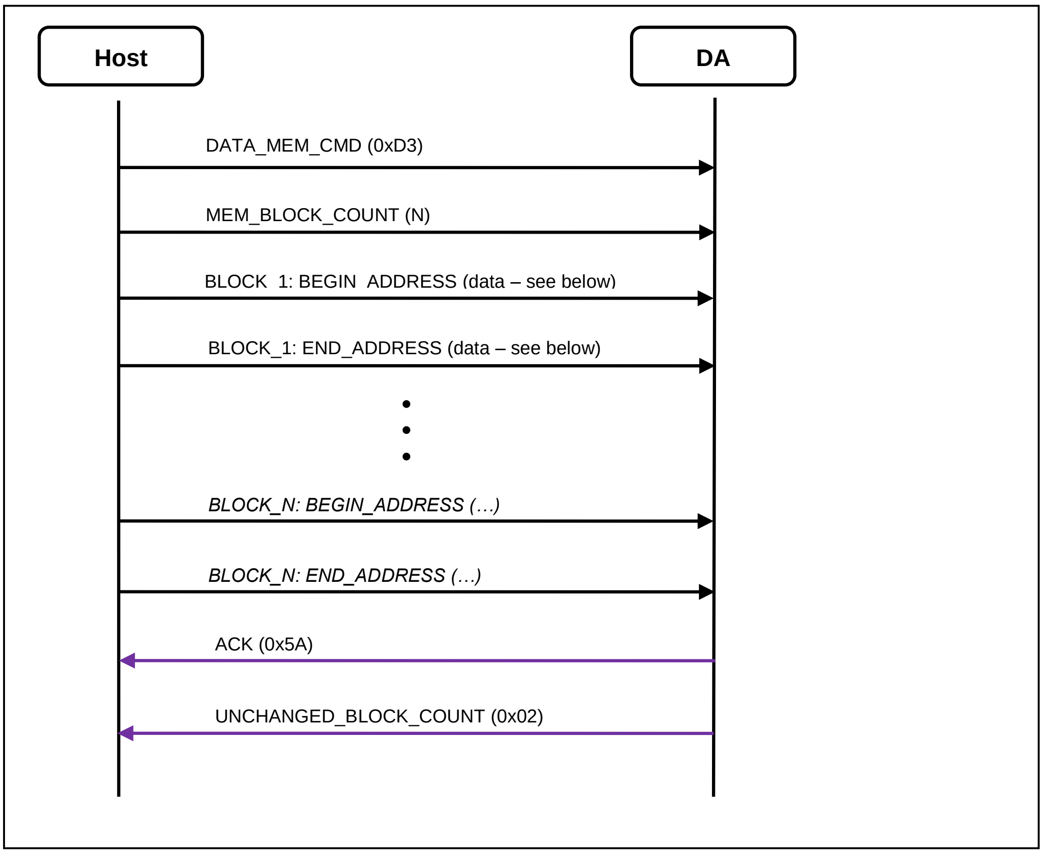
Host
DA
DATA_MEM_CMD (0xD3)
MEM_BLOCK_COUNT (N)
BLOCK_1: BEGIN_ADDRESS (data – see below)
BLOCK_1: END_ADDRESS (data – see below)
BLOCK_N: BEGIN_ADDRESS (…)
BLOCK_N: END_ADDRESS (…)
ACK (0x5A)
UNCHANGED_BLOCK_COUNT (0x02)

Figure 4: Flow of CMD_SetMemBlock

Data arguments:

- MEM_BLOCK_COUNT (N):
- The number of ROM files to be download and the same number of the memory blocks to be allocated by the module. Where the number of the ROM file is 1, then N = 1.
- BLOCK_1: BEGIN_ADDRESS:
- The starting address for the ROM file in memory, for the block 1.
- BLOCK_1: END_ADDRESS:
- The end address for the ROM file in memory, for the block 1.
- According to the protocol requirements, the ROM file is sent in packets with the fixed length of 256 bytes (plus additional checksum of 2 bytes). When there is not enough bytes of data to be sent, such as the last packet at the end of the file (the remaining bytes < 256), the downloading shall fill padding zeros for the rest of the packet.
- As a result of the requirement, the END_ADDRSS of a memory block must include the additional padding bytes in size.

- Example: If the ROM file size is 500324 bytes long, there expected a length of 156 bytes padding bytes being appended to the remaining bytes of the ROM file in the last packet.
- END_ADDRSS shall be the total size – 1.

3.4. Download BIN (ROM) File

3.4.1. Command: CMD_WriteData

This command is used to write all the data of BIN files to target side flash.

- Every packet is fixed length; that is PACKET_LENGTH plus two bytes checksum.
- If DA successfully received this packet, it will return CONT_CHAR (0x69) to notify BROM DLL continues to send the next packet.
- As stated in the section 3.3.3 Command: CMD_SetMemBlock, the last packet is usually not enough for PACKET_LENGTH, the code shall fill 0x00 as padding until reach PACKET_LENGTH.
- At the last, DA will perform checksum of all the BIN files from flash and compare with the checksum from UART. If both checksum are the same, DA will return ACK (0x5A) that means they are successfully written to flash, otherwise, return NACK (0xA5) to indicate errors.

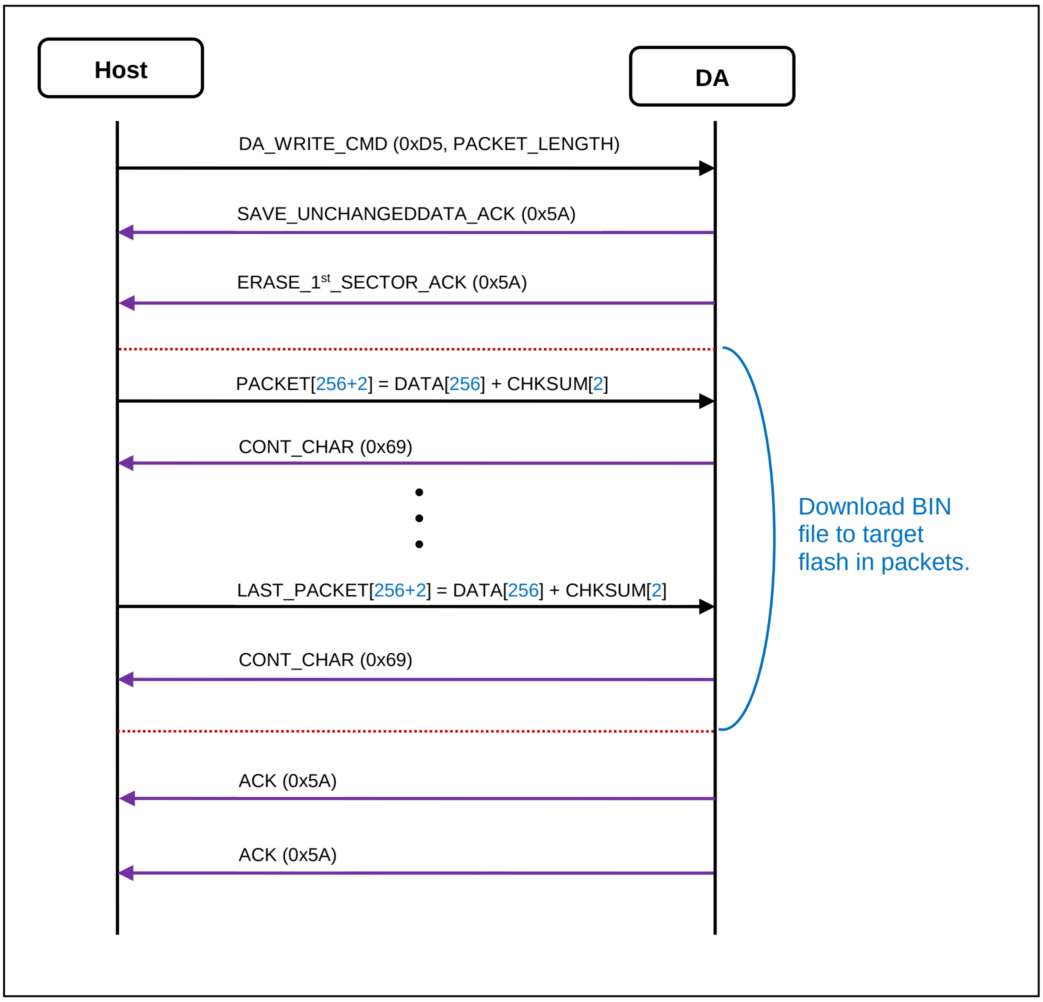
Host
DA
DA_WRITE_CMD (0xD5, PACKET_LENGTH)
SAVE_UNCHANGEDDATA_ACK (0x5A)
ERASE_1st_SECTOR_ACK (0x5A)
PACKET[256+2] = DATA[256] + CHKSUM[2]
CONT_CHAR (0x69)
LAST_PACKET[256+2] = DATA[256] + CHKSUM[2]
CONT_CHAR (0x69)
Download BIN file to target flash in packets.
ACK (0x5A)
ACK (0x5A)

Figure 5: Download BIN file - flow of CMD_WriteData

Data:

- For PACKET_LENGTH = 256:

DA_WRITE_CMD (0xD5 0x00 0x00 0x01 0x00)

3.4.2. Command: CMD_Finish

This command is used to notify DA to power off target by unlock RTC power key.

The firmware update procedure is thus complete and the GNSS module will restart automatically.

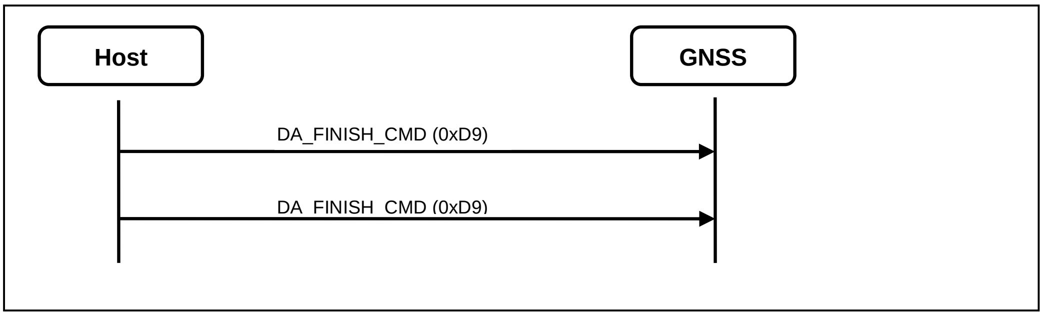

Figure 6: Download BIN file - CMD_Finish

4. Document History

Revision	Date	Changes
0	2014-12-20	Telit Jupiter MT33xx Host Flash Application Note
1	2015-03-23	Added messages that were missing in the previous edition and more details in the descriptions.
2	2015-04-02	1) Added SC872-A module to the applicability table. 2) Added captions to the figures as well as the contents to the Contents of Figure. 3) Some fond type changes for consistency