Wi-SUN Enhanced HAN
Plus Route-B Dual Stack
J11 OTA Update Specification

First Edition

English

ISB CORPORATION

Notice

1. The contents of this document are the latest at the time of the publication of this document and may be subject to change without notice.
2. ISB Corporation does not guarantee that there are no errors in the information. Even in the event that any damage or loss arising from any and all errors in the information provided in this document is caused to you, ISB Corporation shall not be responsible whatsoever for such errors.
3. ISB Corporation shall not be responsible whatsoever for any and all third-party infringements of patents, copyrights, and other intellectual property rights that were caused in relation to the use of technical information provided with this document. ISB Corporation shall, in accordance with this document, not grant any and all rights based on ISB Corporation's or third party's patents, copyrights, and other intellectual property rights.
4. Reproducing or copying this document, in whole or in part, is strictly prohibited without advance permission of ISB Corporation.

Document Convention

This document uses the following typographical convention:

Convention	Description
Request OTA Data Write	Words in bold with the first letter of each word capitalized indicate command names.

Revision History

Date	Description
March 19, 2020	First English edition of the Japanese original first edition (1.0)
June 1, 2020	Revision of the first English edition

Copyright

Contents

List of Figures

List of Tables

1. Introduction

1.1 Overview

This document describes the firmware update function using Wi-SUN communication on ISB-made Wi-SUN stack (hereinafter referred to as the "Wi-SUN Dual Stack") that is embedded in wireless modules compliant with Wi-SUN Profile for ECHONET Lite as the international wireless communications standards (hereinafter referred to as the "Wi-SUN") for Route B and for Enhanced HAN (hereinafter referred to as the "HAN") specified by "Wi-SUN Alliance" (hereinafter referred to as the "Module").

1.2 Scope of this specification

This document describes the functional specification on OTA update, one of the functions included in Wi-SUN Dual Stack.

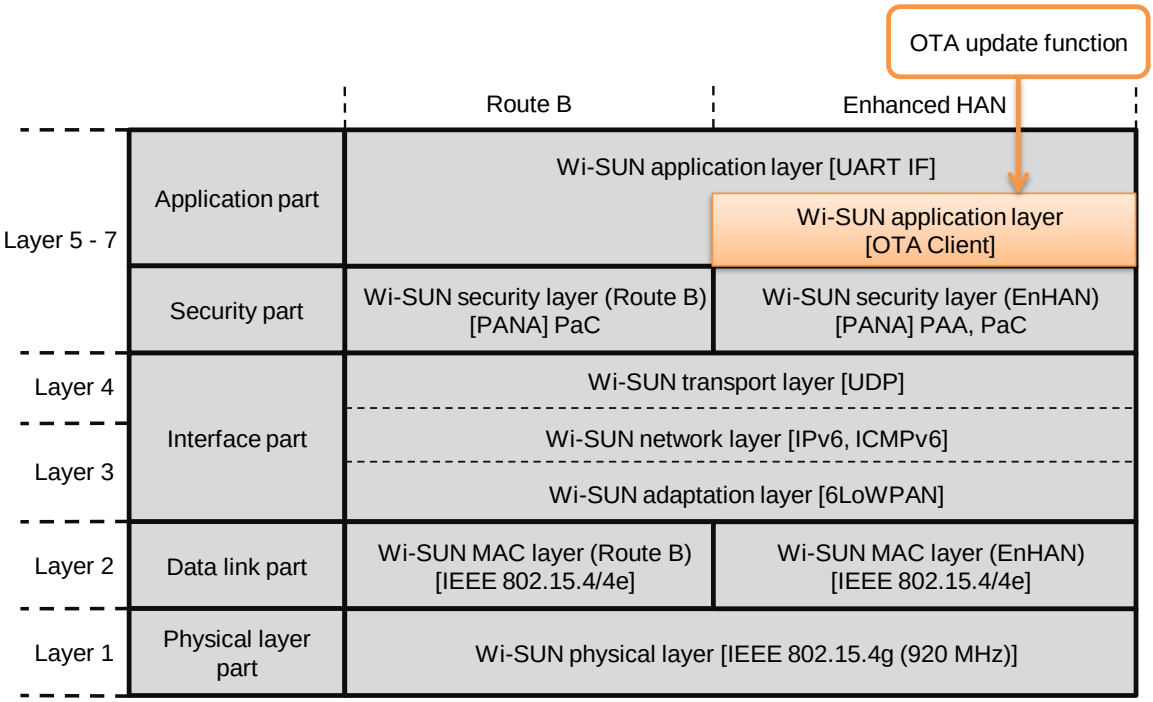

Fig. 1: Scope of this document

1.3 Terms and definitions

Table 1: Terms and definitions

Term	Definition
Route B	Wi-SUN profile for communications between smart meters and HEMS controllers
Enhanced HAN	Wi-SUN profile for communications between HEMS controllers and home electronics
OTA	Over The Air
ISP	Abbreviation of In-System Programming. Function to rewrite on-chip flash ROM that is embedded in a system.

1.4 Reference documents

Table 2: Reference documents

No.	Document name
1	J11 UART IF Specification

2. OTA update function

2.1 Overview

The OTA update function is firmware update function by OTA update communication using Wi-SUN communication. OTA update communication uses UDP/IP communication and a form of OTA Server to update firmware and OTA Client to have its firmware updated.

The figure below shows an example case of “firmware update of OTA update from a gateway” and “OTA update from an OTA update dedicated device (node)”.

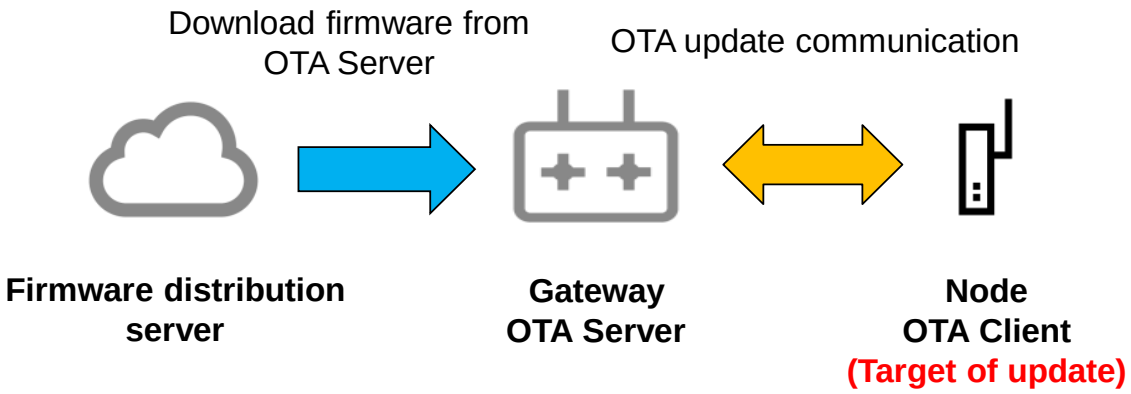

Fig. 2: OTA update from gateway router

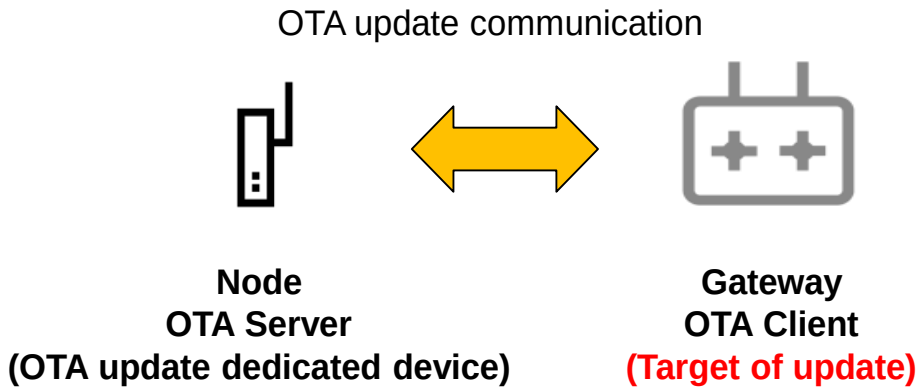

Fig. 3: OTA update from OTA update dedicated device

2.2 Features

- OTA Server updates OTA Client's firmware by transmitting UDP packet specified in this document. Thus, Wi-SUN HAN stack with encrypted UDP communication allows you to use OTA Client's OTA update function.
- OTA update communication provides high reliable firmware update with authentication and encryption.
- Firmware update is written into a different BANK from the currently running BANK; therefore, even if power interruption or other trouble occurs during OTA update, the firmware can recover to previously running firmware.
- OTA update function is designed to prevent from updating firmware for other than Wi-SUN Dual Stack.
- OTA update writing requires approximately 3 minutes for completion. The writing completion time varies, depending on wireless communication circumstance and OTA Server operation.
- After OTA update is completed, reboot of the Module is required to apply the firmware.
  For details of sequence to apply the update, see Chapter 6, "Sequence to apply firmware after rewrite".

2.3 Target of update

The target of OTA update is Wi-SUN Dual Stack firmware in the wireless Module.

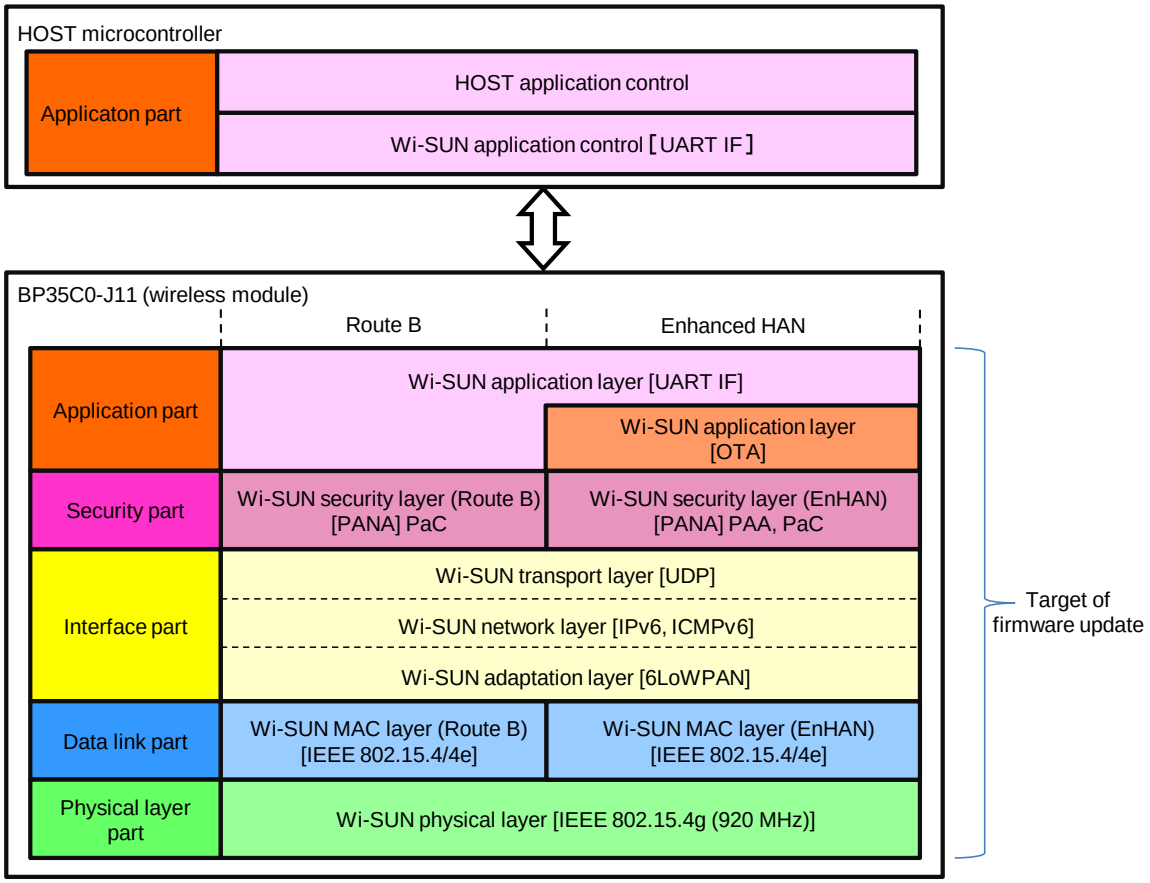

Fig. 4: Target of OTA update

2.4 BANK management

BANK management function is designed to rewrite and activate two BANKs alternately. When updating firmware by OTA update, writing is executed into BANK1 when BANK0 is active or into BANK0 when BANK1 is active.

Two different firmware files are provided for BANK0 and BANK1 update, respectively.

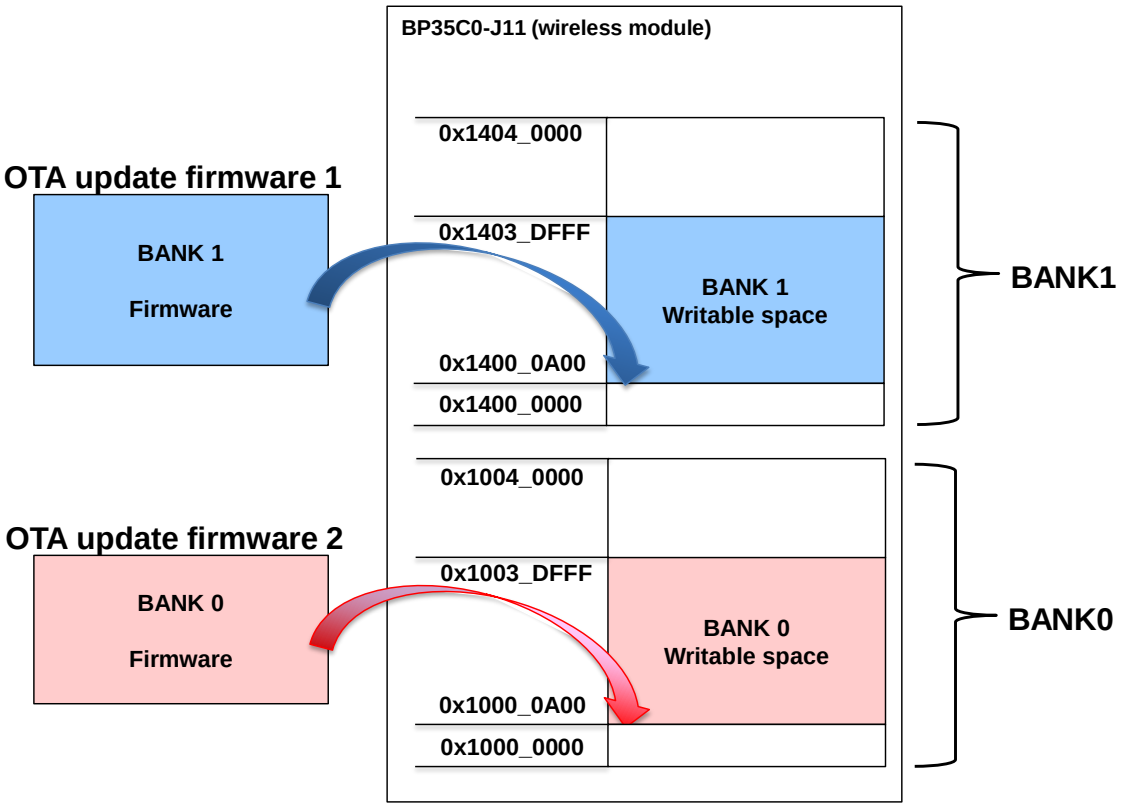

Fig. 5: BANK management

2.5 Prevention of unauthorized access

For prevention of unauthorized access, unauthorized access is determined by checking timing of firmware writing and firmware start-up.

- Determination of unauthorized access at firmware writing
  When firmware for other than Wi-SUN Dual Stack firmware is written by OTA update, an error occurs and Wi-SUN Dual Stack determines that the firmware is invalid.
- Determination of unauthorized access at firmware activation
  Wi-SUN Dual Stack determines whether the firmware is running in BP35C0-J11 when the firmware is activated, and prevents the activation when an error occurs.

3. OTA update communication

OTA update communication means UDP/IP communication to control the OTA update function in the Module.

3.1 Overview

- OTA update communication provides UDP/IP communication using the port number 31941.
- OTA update communication defines OTA Server to update firmware and OTA Client to have its firmware updated.
- When OTA Client requests **Initiate OTA Client** (§4.3.1.1), the UDP port number 31941 is automatically open.
- In order for OTA Server to receive a response from OTA Client, UDP port number 31941 should be open by the upper-level application.
- OTA update communication requires communication after connection establishment with Wi-SUN.
- OTA update communication does not accept plain text data. Encryption with Wi-SUN is required.
- OTA update function is single task. Therefore, while OTA process is being executed (waiting for a response to a request), further request data cannot be processed.
- Checksum is used to verify if there are any errors in received UDP data.
- It is necessary to implement the upper-level application of OTA update communication into OTA Server in order for OTA Server to perform OTA update communication with OTA Client and check responses from OTA Client.
- Retransmission at failure of OTA Server's OTA update communication should be implemented by the upper-level application.
- When OTA Client received UDP packets for OTA update toward OTA port after initiating OTA Client, OTA Client automatically analyzes UDP packets for OTA update and performs internal process according to the requested data. OTA Client responds the processed results using UDP packets for OTA update.
- The maximum delay time of **Response** command is 10 seconds.

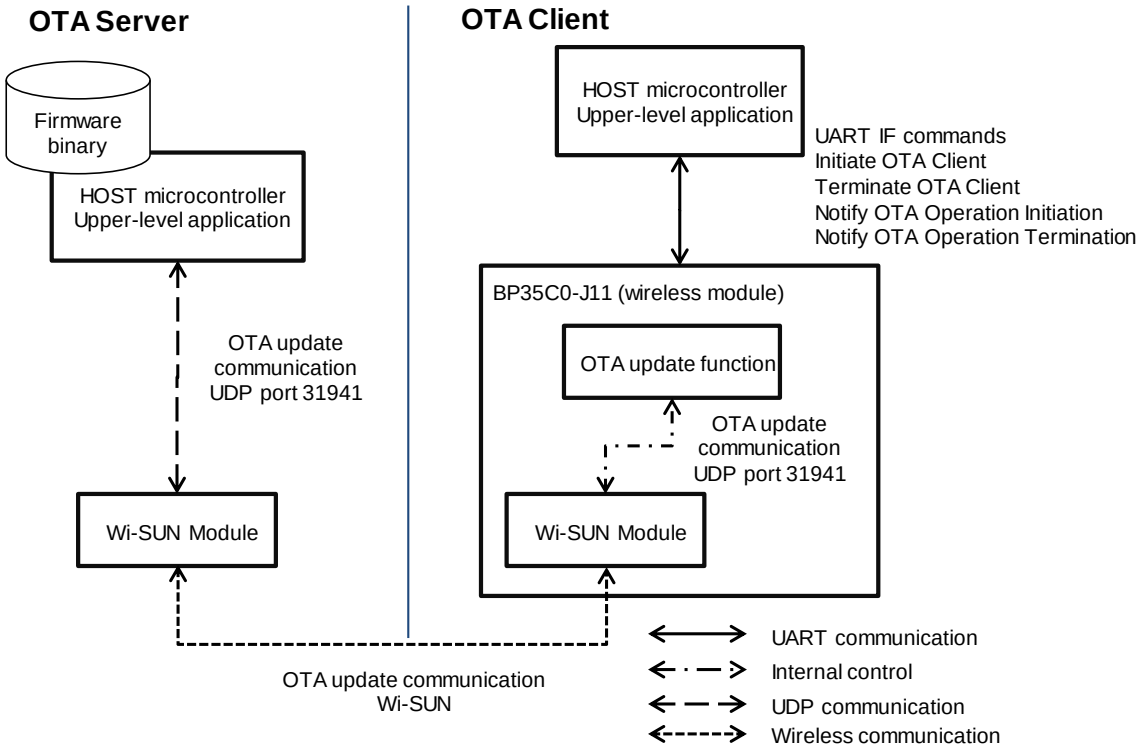

Fig. 6: OTA update communication system

3.2 OTA UDP packet format

OTA UDP packet falls into 2 types of OTA control packet and OTA write packet.

Set the following format into the UDP payload.

3.2.1 OTA control packet format

Control packet header (1byte)	Message length (1byte)	Command code (1byte)	Command parameter (variable)	Checksum (1byte)	Control packet footer (1byte)

Fig. 7: OTA control packet format

Table 3: OTA control packet format

Name	Description	Value
Control packet header	Identifier of OTA control packet	0x01
Message length	Length of command code + command parameter	-
Command code	See Table 4: Command codes.	-
Command parameter	See §4.2.1 “Request/Response commands (control)”.	-
Checksum	Checksum of message length + command code + command parameter	-
Control packet footer	End of OTA control packet	0x03

3.2.1.1 Command codes

OTA control packet is classified into two types: **Request** command used to make a request from OTA Server to OTA Client and **Response** command used to make a response to the request from OTA Client to OTA Server.

Table 4: Command codes

Command name	Command code	
	Request	Response
Start OTA Write	0x40	0x70
End OTA Write	0x45	0x75
Start OTA Mode	0x60	0x71
Get OTA Write BANK Information	0x62	0x72
End OTA Mode	0x64	0x74
Get OTA Version Information	0x68	0x78

3.2.2 OTA write packet format

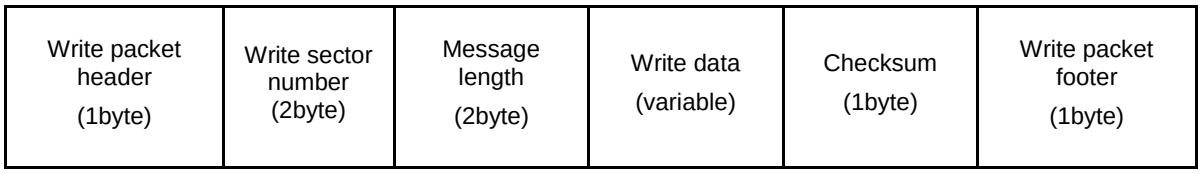

Fig. 8: OTA write packet format

Table 5: OTA write packet format

Name	Description	Value
Write packet header	Identifier of OTA write packet identifier	0x02
Write sector number	Relative sector number of write data	-
Message length	Length of write data	-
Write data	Request: Firmware write data. See §4.2.2.1.1. Response: Firmware write result. See §4.2.2.2.1.	-
Checksum	Checksum of write sector number + message length + write data	-
Write packet footer	End of OTA write packet: 0x03(ETX) Transmission continued of OTA write packet: 0x17(ETB)	0x03/0x17

3.3 Creating OTA write packet

Firmware of Wi-SUN Dual Stack is provided in the Intel Hex format. The padding of space that is out of range of the firmware is 0xFF.

To create OTA write packet, firmware data including the start address is divided by the sector size (512 bytes). The write sector number begins at 1 that contains 512 bytes from the start address, and the following sector numbers are assigned sequentially to the relative sector numbers.

The Figure below shows an example of dividing firmware by 512 bytes.

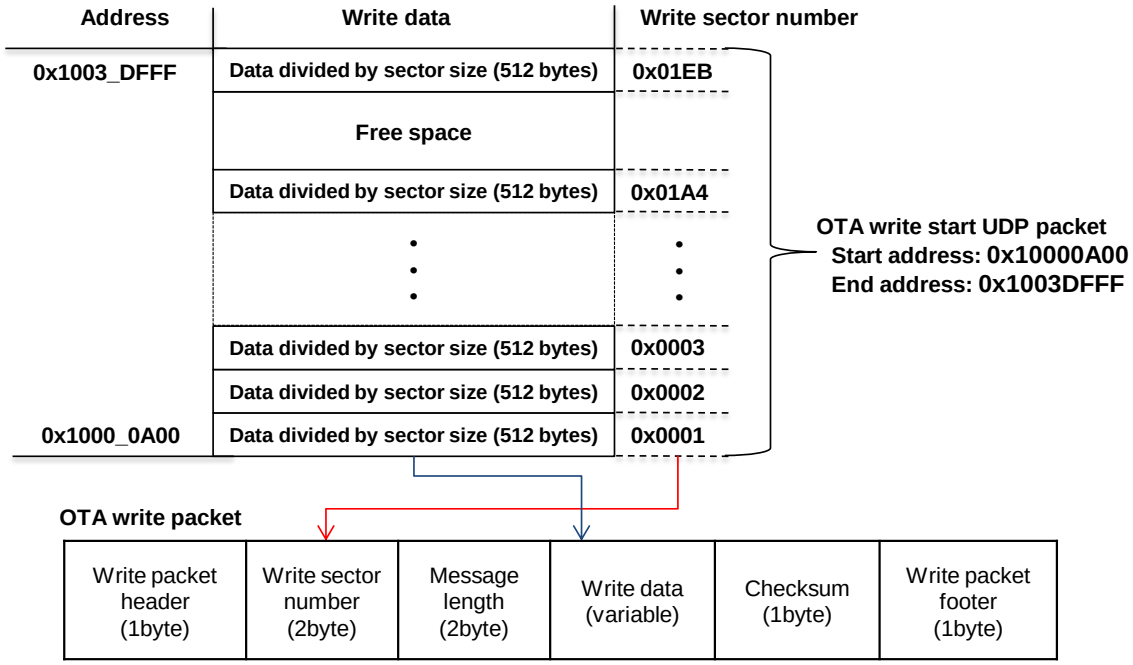

Fig. 9: Example of dividing firmware (BANK0)

OTA Client that received OTA write packet calculates an address where data will be written by using the start address specified by OTA write start packet and write sector number (write destination address = start address + (write sector number - 1) x 512 bytes). Note that the divided data must match the write sector number.

The firmware does not include space (continuous 0xFF) that does not need data writing. When the firmware data are merely connected, address information included in the Intel Hex format is lost. To prevent this, create OTA write packet matching the address specified in the Intel Hex format and address of individual divided data.

When free space exists in write data space, there is no necessary to transmit OTA write packet as large as the free space. In case of the above example, after 0x01A4 OTA write packet transmission, the following 0x01D0 OTA write packet can be transmitted.

Specify a multiple of 4 since the minimum write unit of a message length is 1 word (4 bytes).

For the end of OTA write packet (when write sector number = 0x01D0 in the above example), specify 0x03(ETX) in write packet footer.

When OTA write packet is anything but the end, specify 0x17(ETB) in write packet footer.

3.4 Checksum

Error detection is performed on OTA UDP packet using checksum. If value integrity check is invalid, **Respond Error** is executed.

The value decremented by 1 byte each from all data starting at the initial value 0x00 (borrow is ignored).

3.4.1 OTA control packet checksum

Target of calculation: Message length, command code, command parameter

Calculation example:

For **Start OTA Mode Request**:

Checksum = 0x00(initial value) - 0x01(message length) - 0x61(command code) - (none) (command parameter)

In case of the above example, “0x9E” is the checksum value.

3.4.2 OTA write packet checksum

Target of calculation: Write sector number, message length, write data

Calculation example:

Checksum = 0x00(initial value) - 0x00(write sector number 1) - 0x01(write sector number 2) - 0x00(message length 1) - 0x04(message length 2) - 0xFF (write data 1) - 0x80 (write data 2) - 0x40 (write data 3) - 0x22 (write data 4)

In case of the above example, “0x1A” is the checksum value.

3.5 OTA Client status transition

The diagram below shows the status transition of OTA Client.

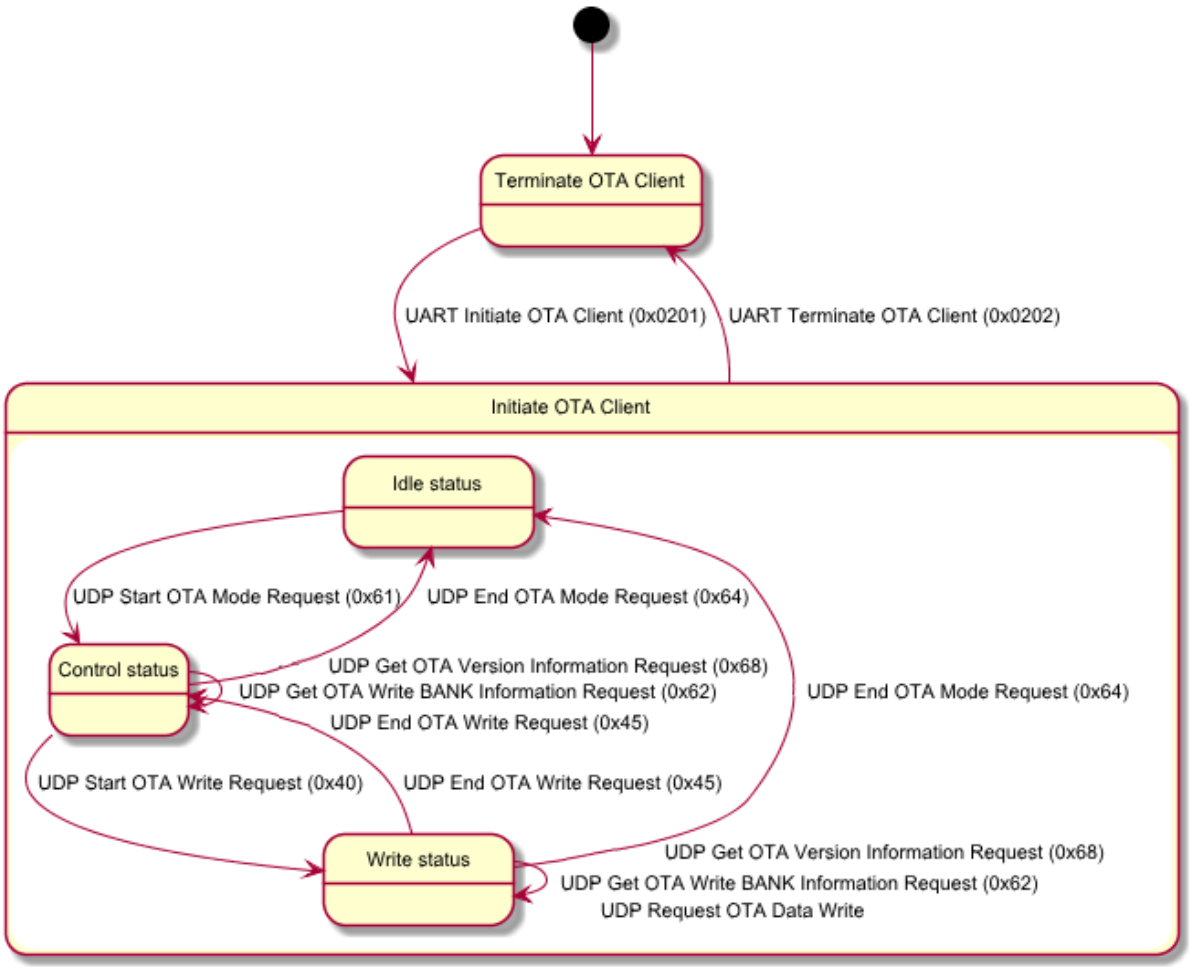

Fig. 10: OTA Client status transition diagram

Table 6: OTA Client status

Status description	Description
Idle status	Status where OTA Client starts to accept Start OTA Mode.
Control status	Status where OTA mode starts to receive any OTA control packet.
Write status	Status where OTA write starts to receive OTA write packet.

4. OTA update IF specification

4.1 OTA update IF

4.1.1 OTA UDF packet

The table below lists commands to use UDP packet in the OTA update function.

Table 7: UDP packet commands

Major category	Classification	Command name	Command code	
			Request	Response
UDP	Control	**Start OTA Write**	0x40	0x70
		End OTA Write	0x45	0x75
		Start OTA Mode	0x61	0x71
		Get OTA Write BANK Information	0x62	0x72
		End OTA Mode	0x64	0x74
		Get OTA Version Information	0x68	0x78
	Write	**Request OTA Data Write**	NA	NA
		Respond OTA Data Write	NA	NA
	Other	**Respond Error**	NA	0xE0

4.1.2 UART IF commands

The table below lists commands to use UART IF in the OTA update function.

Table 8: UART IF commands

Major category	Classification	Command name	Command type		
			Request	Response	Notification
UART	Operation	**Initiate OTA Client**	0x0201	0x2201	NA
		Terminate OTA Client	0x0202	0x2202	NA
		Notify OTA Operation Initiation	NA	NA	0x6033
		Notify OTA Operation Termination	NA	NA	0x6034

4.2 OTA UDP packet

4.2.1 Request/Response commands (control)

4.2.1.1 Start OTA Write

Request command	0x40	Response command	0x70
Function description			

To make OTA Client transition to the write status to allow write of firmware program code into a specified address space.

A **Request** with a specified space including active BANK space results in an error.

4.2.1.1.1 Request command parameters

Name	Length in bytes	Range	Detail
Write start address	4	0x10000A00 or 0x14000A00	Start address of write firmware
Write end address	4	0x1003DFFF or 0x1403DFFF	End address of write firmware

4.2.1.1.2 Response command parameter

Name	Length in bytes	Range	Detail
Response result	1	-	For command execution results, see Table 9: List of response results.

4.2.1.2 End OTA Write

Request command	0x45	Response command	0x75
Function description			

To end OTA Client write operation to make the transition to the control status.

This packet is used to notify that all firmware data transmission from OTA Sever to OTA Client was successfully completed.

After receiving this packet, OTA Client checks written firmware, if no error is found, and registers it as the firmware to start next time into the Module.

The registered firmware will take effect next time the Module starts up.

4.2.1.2.1 Request command parameter

Name	Length in bytes	Range	Detail
None	-	-	-

4.2.1.2.2 Response command parameter

Name	Length in bytes	Range	Detail
Response result	1	-	For command execution results, see Table 9: List of response results.

4.2.1.3 Start OTA Mode

Request command	0x61	Response command	0x71
Function description			

To make OTA Client transition to the control mode to receive any OTA control packet.

OTA Client notifies the HOST that OTA update control starts by **Notify OTA Operation Initiation** (§4.3.2.1).

4.2.1.3.1 Request command parameter

Name	Length in bytes	Range	Detail
None	-	-	-

4.2.1.3.2 Response command parameter

Name	Length in bytes	Range	Detail
Response result	1	-	For command execution results, see Table 9: List of response results.

4.2.1.4 Get OTA Write BANK Information

Request command	0x62	Response command	0x72
Function description			

To check an OTA update write target BANK with OTA Client.

OTA Server selects firmware written in the BANK of the number that OTA Client responds accordingly and starts OTA write.

4.2.1.4.1 Request command parameter

Name	Length in bytes	Range	Detail
None	-	-	-

4.2.1.4.2 Response command parameters

Name	Length in bytes	Range	Detail
Response result	1	-	For command execution results, see Table 9: List of response results. Note: In case of failure to receive the command, the following parameters will not be given.
Write BANK number	1	0x00 to 0x01	0x00: BANK0 0x01: BANK1

4.2.1.5 End OTA Mode

Request command	0x64	Response command	0x74
Function description			
To end receiving OTA control packet to make OTA Client transition to the idle status. OTA Client notifies the HOST that OTA update control ended by **Notify OTA Operation Termination** (§4.3.2.2).			

4.2.1.5.1 Request command parameter

Name	Length in bytes	Range	Detail
None	-	-	-

4.2.1.5.2 Response command parameter

Name	Length in bytes	Range	Detail
Response result	1	-	For command execution results, see Table 9: List of response results.

4.2.1.6 Get OTA Version Information

Request command	0x68	Response command	0x78
Function description			
To get the firmware version information on OTA Client.			

4.2.1.6.1 Request command parameter

Name	Length in bytes	Range	Detail
None	-	-	-

4.2.1.6.2 Response command parameters

Name	Length in bytes	Range	Detail
Response result	1	-	For command execution results, see Table 9: List of response results. Note: In case of failure to receive the command, the following parameters will not be given.
Firmware ID	2	-	0x0400: Wi-SUN Enhanced HAN Plus Route-B Dual Stack
Major version	1	0x00 to 0xFF	Current version
Minor version	1	0x00 to 0xFF	Current version
Revision	4	0x00000000 to 0xFFFFFFFF	Current revision

4.2.2 Request/Response commands (write)

4.2.2.1 Request OTA Data Write

Request command	-	Response command	-
Function description			
To request OTA Client to write firmware. After receiving **Request OTA Data Write**, OTA Client calculates a write start address by the following formula, after erasing a write start address' sector, and writes firmware data of a size specified in the message length. The data is written in units of word (4 bytes) and if it is 0xFFFFFFFF, write is omitted. Write start address calculation method: Write start address = write start address of **Start OTA Write** (§4.2.1.1) + (write sector number -1) x 512 (sector size)			

4.2.2.1.1 Request command parameter

Name	Length in bytes	Range	Detail
Write data	4 to 512	-	Write firmware data Specify a multiple of 4 since the minimum write unit is 1 word (4 bytes).

4.2.2.1.2 Example of creating a Request command

For an example of creating a **Request** command, see §3.3 "Creating OTA write packet".

4.2.2.2 Respond OTA Data Write

Request command	-	Response command	-
Function description			
To respond the result of OTA Client data write. After firmware data write, the written sector is read and CRC32 is executed. If the response result or write result is an error, 0 is set in checksum.			

4.2.2.2.1 Response command parameters

Name	Length in bytes	Range	Detail
Response result	1	-	For command execution results, see Table 9: List of response results.
Write result	1	-	For data write results, see Table 9: List of response results.
Checksum	4	-	Result of read and CRC32 calculation of written sectors Note: If the response result or write result is an error, 0 is set.

4.2.3 Response command (other)

4.2.3.1 Respond Error

Request command	-	Response command	0xE0
Function description			
To respond an error response code when an error is found due to undefined command codes and invalid UDP packet format.			

4.2.3.1.1 Response command parameter

Name	Length in bytes	Range	Detail
Response result	1	-	For command execution results, see Table 9: List of response results.

4.2.4 List of response results

The table below lists response results set to **Response** commands.

Table 9: List of response results

Response result (DEC)	Response result (HEX)	Description
5	0x05	Error with invalid command information (parameter)
6	0x06	Response succeeded.
7	0x07	Error with abnormal frame data received
21	0x15	Negative response (error with failure of unmatch of the status, etc.)
28	0x1C	Write error to flash memory
29	0x1D	Write process was omitted because write data is all "0xFF" or the same data have been written.
30	0x1E	Error of integrity check on the entire firmware

4.3 UART IF commands

4.3.1 Request/Response command (operation)

4.3.1.1 Initiate OTA Client

Request command	0x0201	Response command	0x2201
Function description			
To initiate OTA Client and put it into operation in the status in which OTA UDP packets can be accepted. At execution of **Initiate OTA Client**, open the UDP port of 31941 to be used.			

4.3.1.1.1 Request command parameter

Name	Length in bytes	Range	Detail
None	-	-	-

4.3.1.1.2 Response command parameter

Name	Length in bytes	Range	Detail
Response result	1	-	For command execution results, see the document, “J11 UART IF Specification”.

4.3.1.2 Terminate OTA Client

Request command	0x0202	Response command	0x2202
Function description			
To terminate OTA Client and return it to the status in which no OTA UDP packets can be accepted. At execution of **Terminate OTA Client**, close the UDP port of 31941 in use.			

4.3.1.2.1 Request command parameter

Name	Length in bytes	Range	Detail
None	-	-	-

4.3.1.2.2 Response command parameter

Name	Length in bytes	Range	Detail
Response result	1	-	For command execution results, see the document, “J11 UART IF Specification”.

4.3.2 Notification command (operation)

4.3.2.1 Notify OTA Operation Initiation

Notification command	0x6033
Function description	
To notify the initiation of OTA updating operation. The initiation is notified when the OTA Client receives an OTA mode initiation packet.	

4.3.2.1.1 Notification command parameter

Name	Length in bytes	Range	Detail
Source IPv6 address	16	0xFE80000000000000XXXXXXXXXXXXXXXX XX represents MAC address.	IPv6 address of the source of OTA data Note: The lower 2nd bit of the first 1 byte of the MAC address is inverted.

4.3.2.2 Notify OTA Operation Termination

Notification command	0x6034
Function description	
To notify the termination of OTA updating operation. The termination is notified when the OTA Client receives an OTA mode termination packet.	

4.3.2.2.1 Notification command parameters

Name	Length in bytes	Range	Detail
OTA update result	1	0x01 to 0x03	0x01: Version upgrade succeeded 0x02: Version upgrade failed 0x03: No version upgrade
Source IPv6 address	16	0xFE80000000000000XXXXXXXXXXXXXX XX represents MAC address.	IPv6 address of the source of OTA data Note: The lower 2nd bit of the first 1 byte of the MAC address is inverted.

5. OTA update sequence

This chapter describes sequences between OTA Server and OTA Client after Wi-SUN connection completion.

Firmware can be rewritten by sequentially executing steps 1. to 3. below.

Steps

1. Sequence of OTA update start (see §5.1)
2. Sequence of OTA update write (see §5.2)
3. Sequence of OTA update end (see §5.3)

5.1 Sequence of OTA update start

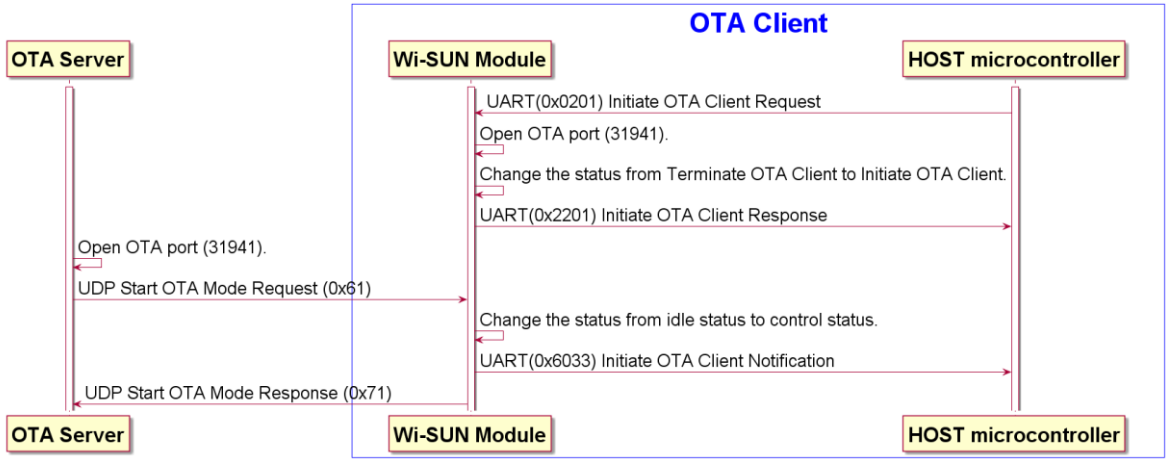

Fig. 11: Sequence of OTA update start

5.2 Sequence of OTA update write

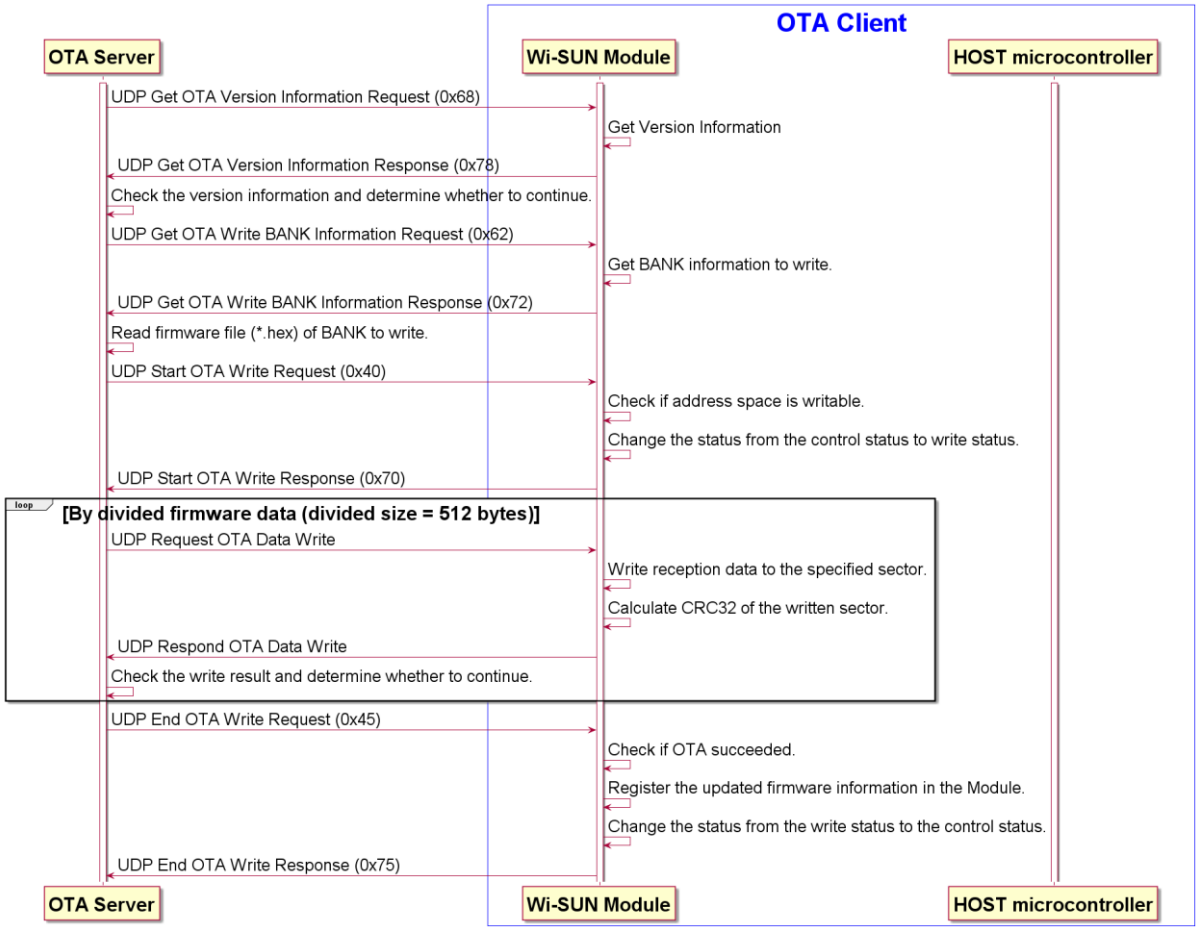

Fig. 12: Sequence of OTA update write

5.3 Sequence of OTA update end

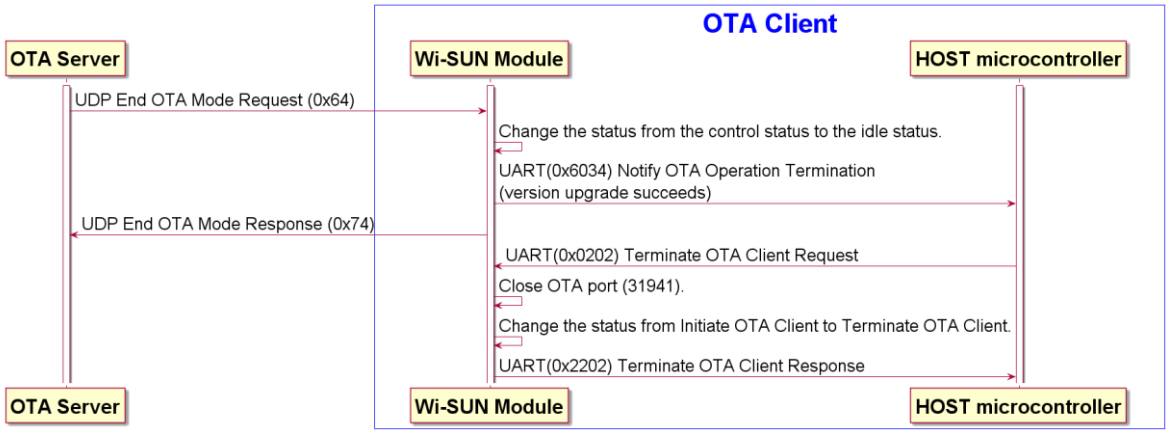

Fig. 13: Sequence of OTA update end

6. Sequence to apply firmware after rewrite

After successful OTA update, to make the update take effect, restart of the Module is required. An example of sequence to apply firmware update is shown below.

The sequence includes the following 2 cases:

- OTA Client leads to apply written firmware
- OTA Server leads to apply written firmware

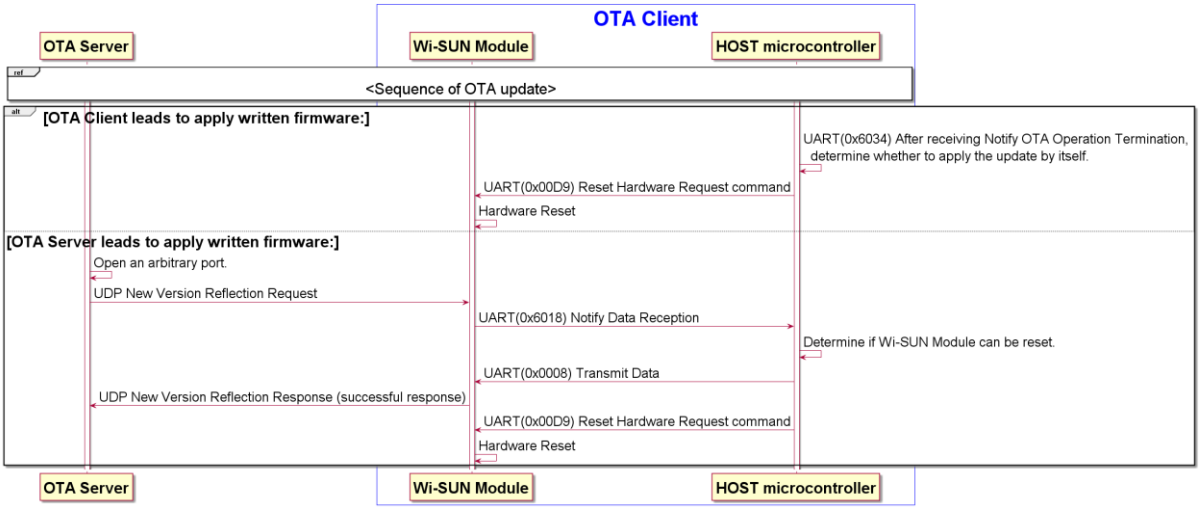

Fig. 14: Sequence to apply rewritten firmware

7. Operation when an error is found

7.1 OTA UDP packet reception during command processing

While OTA Client is being process a command, OTA UDP packet cannot be processed. In this case, **Respond Error** is not executed and all UDP packets received are discarded. After transmitting a **Request** command, wait until receiving a **Response** command, and then transmit a **Request** command.

7.2 Invalid parameter of OTA UDP packet

If an undefined value is specified in a **Request** command's parameter, **Respond Error** is executed and all UDP packets received are discarded.

7.3 Error during firmware write

If an error occurs with firmware write at OTA Client, **Respond Error** is executed and all UDP packets received are discarded.

If an error occurs with **Respond OTA Data Write** (§4.2.2.2) at OTA Server, retransmit **Request OTA Data Write** (§4.2.2.1) where an error occurred.

7.4 Write error of firmware for other than Wi-SUN Dual Stack

When firmware for other than Wi-SUN Dual Stack is written for OTA update, OTA Client detects an error during integrity check process of **End OTA Write** (§4.2.1.2) and transmits **Respond Error.**

7.5 Damage check and rollback at Module start-up

When the Module starts up, damage check of the firmware is performed. If damaged, firmware rollback is executed to revert to the previous firmware version and start the Module.

If the previous firmware version is also damaged, the Module starts up in a firmware rewrite mode via UART.

7.6 Error in writing firmware of a non write target BANK

When OTA Server transmits firmware data in BANK that is not a write target (in other words firmware running in the active BANK), OTA Client detects an error during integrity check process of **End OTA Write** (§4.2.1.2) and transmits **Respond Error**.

Recheck a write target BANK and retransmit correct firmware data by **Request OTA Data Write** (§4.2.2.1).

8. Notes

- The flash memory embedded in BP35C0-J11 can be rewritten up to 100 times; therefore, use the OTA update function only if necessary.
- When OTA Client is an end device with the sleep function enabled. OTA update is executed via indirect communication. Accordingly, periodically transmit a poll request during a period of time from requesting **Initiate OTA Client** (§4.3.1.1) until receiving **Notify OTA Operation Termination** (§4.3.2.2).
- OTA Client executes OTA update by following packets transmitted from OTA Server; therefore, exclusively control OTA update by the upper-level application in order to avoid OTA update execution from multiple OTA Servers to the same OTA Client.