Wi-SUN Enhanced HAN
Plus Route-B Dual Stack
J11 OTA Update Sample Scripts
Operation Manual

Edition 1.1

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.

5. ISB Corporation shall not be held liable or responsible in any way for any direct or indirect damages (including but not limited to damages for information, data, programs or other intangible property, loss of profits, interruption of business) that may occur to the customer as a result of the use and/or inability to use sample programs described in this document.

Document 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
June 23, 2020	Replacing images of the execution screen

Copyright

Contents

List of Figures

List of Tables

1. Introduction

1.1 Overview

OTA update sample scripts allow you to easily execute the OTA update function embedded in ISB-made Wi-SUN stack and check the operation.

This document describes how to operate OTA update sample scripts.

To execute OTA update using the OTA update function embedded in Wi-SUN stack, OTA Server and OTA Client applications are required to control Wi-SUN stack.

Sample scripts implement the application operation of OTA Server and OTA Client respectively and can be used as a reference for practically developing the application by modifying the scripts and checking behavior variation.

1.2 Configuration

OTA update sample scripts run on the device configuration as shown below.
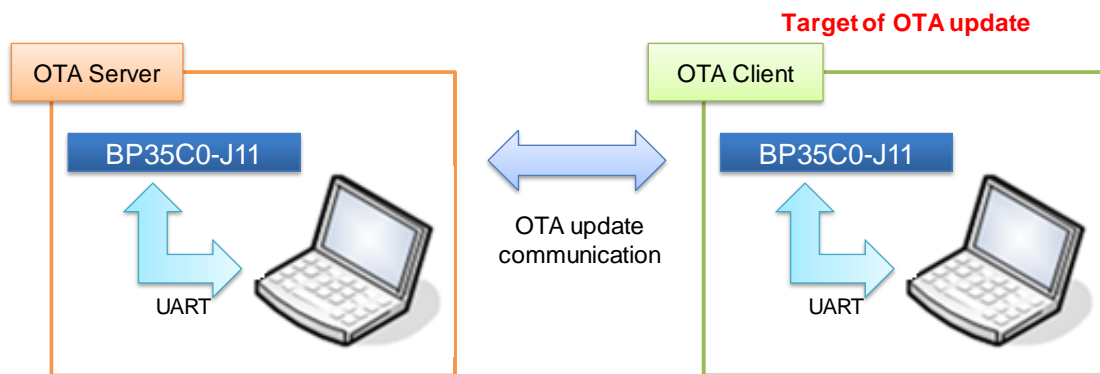
Fig. 1 Configuration diagram

Details of the device configuration shown in Fig. 1 are provided below.

Table 1: Details of hardware used

Hardware	Description
Test PC	PC is used to execute scripts.
	Ubuntu 16.04 LTS desktop 64-bit is recommended.
BP35C0-J11	Wi-SUN wireless module
	Scripts change the operation to the following each mode:
	- OTA Server, HAN PAN coordinator
	- OTA Client, HAN end device

1.3   Terms and definitions

Table 2: Terms and definitions

Term	Definition
Enhanced HAN	Abbreviation of Enhanced Home Area Network Wi-SUN profile for communications between HEMS and home electronics
OTA	Over The Air
OTA Server	Device that controls OTA update
OTA Client	Device that is the target of OTA update

1.4   Reference documents

Table 3: Reference documents

No.	Document name
1	J11 UART IF Specification
2	J11 OTA Update Specification

2. Execution environment

To run sample scripts, the following environment is required for test PC.

- python 3.5 or later is installed.
  python package as described later is installed.
- Wi-SUN wireless module (BP35C0-J11) serial device is recognized.

Verified environment

- Ubuntu 16.04 LTS desktop 64-bit
- Python 3.5.2

2.1 Required Python package

The sample scripts use the following packages.

Before executing the sample scripts, install the packages below.

- pySerial
- IntelHex

Installation instructions

1. Install pip (not necessary if installed).

```
$ sudo apt-get install python3-pip
```

2. Install the package.

```
$ python3 -m pip install pyserial intelhex
```

3. OTA update execution instructions

OTA update instructions using the sample scripts are provided below.

Steps

1. Start up OTA Server.
   For details, see §3.1 "Starting up OTA Server".

2. Start up OTA Client.
   For details, see §3.2 "Starting up OTA Client".

3. Sequentially execute the OTA Update Menu on OTA Server.
   For details, see §3.3 "Sequentially executing OTA Update Menu on OTA Server".

4. Check the update result on OTA Client.
   For details, see §3.4 "Checking update result on OTA Client".

Parameters in execution example of this Chapter are described in the table below.

Table 4: Parameters in execution example

Parameter name		Value
OTA Server	MAC address	00:1D:12:91:00:00:06:0B
	Serial communication device file	/dev/ttyUSB0
OTA Client	MAC address	00:1D:12:91:00:00:07:D9
	Serial communication device file	/dev/ttyUSB1
Channel number		14 (926.5MHz)
PAN ID		0x5555(21845)
HAN PANA authentication password		Char:0123456789ABCDEF
		Hex:303132333435363738394142434 44546

3.1   Starting up OTA Server

Execute sample_server.py.

For details of the command option, see §4.1 "Command option of sample_server.py".

Execution example:

```
$ sudo python3 sample_server.py -s /dev/ttyUSB0 -c 14 -p 5555 -ca 001D12910000088A
-cp 30313233343536373839414243444546
```

Note: "-ca" is the MAC address of OTA Client.

After OTA Server starts up as a HAN PAN coordinator, it waits for HAN connection completion from OTA Client.

The screen after the start-up is shown below.

```
2020/06/23 11:47:08.384 Tx: D0EA83FC 00D9 0004 0416 0000
2020/06/23 11:47:08.654 Rx: D0F9EE5D 6019 0004 0391 0000
Reset Hardware: Success
2020/06/23 11:47:08.654 Tx: D0EA83FC 005F 0008 03A0 000F 01000E00
2020/06/23 11:47:08.670 Rx: D0F9EE5D 205F 0005 0398 0001 01
Set Initial Settings:Success
2020/06/23 11:47:08.670 Tx: D0EA83FC 000A 0006 0349 00AA 5555
2020/06/23 11:47:08.685 Rx: D0F9EE5D 200A 0008 0346 00B9 010E5555
Initiate HAN Operation:Success
2020/06/23 11:47:08.685 Tx: D0EA83FC 002C 001C 0381 04F4 001D12910000088A3031323334353637383
9414243444546
2020/06/23 11:47:08.701 Rx: D0F9EE5D 202C 0005 0365 0001 01
Set HAN PANA Authentication Information:Success
2020/06/23 11:47:08.701 Tx: D0EA83FC 003A 0004 0377 0000
2020/06/23 11:47:08.717 Rx: D0F9EE5D 203A 0005 0373 0001 01
Initiate HAN PANA:Success.
2020/06/23 11:47:08.717 Tx: D0EA83FC 0025 0005 0363 0001 01
2020/06/23 11:47:08.733 Rx: D0F9EE5D 2025 0005 035E 0001 01
Switch HAN Acceptance Connection Mode:Success

PAN coordinator started.
Waiting for client connection...
```

Fig. 2: sample_server.py execution screen (start-up)

3.2   Starting up OTA Client

Execute sample_client.py.

For details of the command option, see §4.2 "Command option of sample_client.py".

Execution example:

```
$ sudo python3 sample_client.py -s /dev/ttyUSB1 -c 14 -sa 001D12910000060B -cp
303132333435363738393941424344546
```

Note: "-sa" is the MAC address of OTA Client.

After OTA Client starts up as a HAN End Device, it waits for completion of HAN connection with OTA Server.

When HAN connection is completed, OTA Client waits for Notify OTA Operation Initiation and Notify OTA Operation Termination.

```
 2020/06/23 11:47:13.002 Tx: D0EA83FC 00D9 0004 0416 0000
 2020/06/23 11:47:13.272 Rx: D0F9EE5D 6019 0004 0391 0000
Reset Hardware: Success
 2020/06/23 11:47:13.272 Tx: D0EA83FC 005F 0008 03A0 0011 03000E00
 2020/06/23 11:47:13.290 Rx: D0F9EE5D 205F 0005 0398 0001 01
Set Initial Settings:Success
 2020/06/23 11:47:13.290 Tx: D0EA83FC 000A 000C 034F 07F8 FFFFFFFFFFFFFFFF
 2020/06/23 11:47:15.868 Rx: D0F9EE5D 200A 0011 034F 0268 010E5555001D12910000060BDE
Initiate HAN Operation:Success
 MAC Connected to '001D12910000060B', rssi=-34dBm
 2020/06/23 11:47:15.868 Tx: D0EA83FC 002C 0014 0379 03A2 303132333435363738393941424344546
 2020/06/23 11:47:15.881 Rx: D0F9EE5D 202C 0005 0365 0001 01
Set HAN PANA Authentication Information:Success
 2020/06/23 11:47:15.882 Tx: D0EA83FC 003A 0004 0377 0000
 2020/06/23 11:47:15.914 Rx: D0F9EE5D 203A 000D 037B 00D2 01001D12910000060B
Initiate HAN PANA:Success.
 PANA auth started with '001D12910000060B'

End Device started.
Waiting for authentication completion...
 2020/06/23 11:47:16.201 Rx: D0F9EE5D 6028 000D 03A9 00D2 01001D12910000060B
PANA Auth Complete '001D12910000060B'
 2020/06/23 11:47:16.202 Tx: D0EA83FC 0201 0004 0340 0000
 2020/06/23 11:47:16.217 Rx: D0F9EE5D 2201 0005 033C 0001 01
Initiate OTA Client:Success

Waiting for OTA Update Completion...
```

Fig. 3: sample_client.py execution screen (HAN connection completion)

Notes:

When OTA Client starts up without starting up OTA Server, the start-up fails in the initiation of HAN operation.

After starting up OTA Server, re-start up OTA Client.

```
 2020/06/23 11:23:50.808 Tx: D0EA83FC 00D9 0004 0416 0000
 2020/06/23 11:23:51.074 Rx: D0F9EE5D 6019 0004 0391 0000
Reset Hardware: Success
 2020/06/23 11:23:51.075 Tx: D0EA83FC 005F 0008 03A0 0011 03000E00
 2020/06/23 11:23:51.090 Rx: D0F9EE5D 205F 0005 0398 0001 01
Set Initial Settings:Success
 2020/06/23 11:23:51.090 Tx: D0EA83FC 000A 000C 034F 07F8 FFFFFFFFFFFFFFFF
 2020/06/23 11:23:53.668 Rx: D0F9EE5D 200A 0005 0343 000E 0E
Initiate HAN Operation:Failure
```

Fig. 4: sample_client.py execution screen (HAN connection failure)

3.3   Sequentially executing OTA Update Menu on OTA Server

When OTA Server completes HAN connection with OTA Client, OTA Update Menu appears.

Then, sequentially execute 0. to 7. of OTA Update Menu.

When 04. Start OTA Write is executed, "Please input file path" appears, and enter the file path of firmware (*.hex) for OTA update, and then press Enter.

```
 2020/06/23 11:47:13.311 Rx: D0F9EE5D 601A 000E 039C 0231 01001D12910000088ADE
Notify MAC connection '001D12910000088A'
 2020/06/23 11:47:16.193 Rx: D0F9EE5D 601A 000E 039C 0232 02001D12910000088ADE
Notify PANA auth completion '001D12910000088A'
 2020/06/23 11:47:16.193 Tx: D0EA83FC 0025 0005 0363 0002 02
 2020/06/23 11:47:16.208 Rx: D0F9EE5D 2025 0005 035E 0001 01
Switch HAN Acceptance Connection Mode:Success

OTA Update Menu:
 Please input index
        0. Open UDP Port(31941)
        1. Start OTA Mode
        2. Get OTA Version Information
        3. Get OTA Write BANK Information
        4. Start OTA Write
        5. OTA Data Write
        6. End OTA Write
        7. End OTA Mode
        h. show this Help
>>> 
```

Fig. 5: sample_server.py execution screen (connection completion)

3.3.1 OTA Update Menu execution screen

Execution screens (as examples) with each menu selected are shown below.

```
>>> 0
 2020/06/23 11:49:03.085 Tx: D0EA83FC 0005 0006 0344 0141 7CC5
 2020/06/23 11:49:03.096 Rx: D0F9EE5D 2005 0005 033E 0001 01
Open UDP Port(31941):Success
```

Fig. 6: sample_server.py execution screen (0. Open UDP Port)

```
>>> 1
 2020/06/23 11:49:04.578 Tx: D0EA83FC 0008 001F 0360 065D FE80000000000000021D12910000088A7CC
57CC500050101619E03
 2020/06/23 11:49:04.619 Rx: D0F9EE5D 2008 000B 0347 0105 01000101619E03
Start OTA Mode Request [SEND]:Success
 2020/06/23 11:49:04.649 Rx: D0F9EE5D 6018 0025 03B1 07E8 FE80000000000000021D12910000088A7CC
57CC555550002DE0006010271068703
 OTA Update Packet: 01 02 71 06 87 03
Start OTA Mode Response [RECV:Success]
```

Fig. 7: sample_server.py execution screen (1. Start OTA Mode)

```
>>> 2
 2020/06/23 11:49:06.273 Tx: D0EA83FC 0008 001F 0360 065D FE80000000000000021D12910000088A7CC
57CC500050101689703
 2020/06/23 11:49:06.313 Rx: D0F9EE5D 2008 000B 0347 0105 01000101689703
Get OTA Version Information Request [SEND]:Success
 2020/06/23 11:49:06.332 Rx: D0F9EE5D 6018 002D 03B9 08F0 FE80000000000000021D12910000088A7CC
57CC555550002DE000E010A78060400010300000158AD103
 OTA Update Packet: 01 0A 78 06040001030000158A D1 03
Get OTA Version Information Response [RECV:Success]
 Firmware Version:0400-01-03-0000158A
```

Fig. 8: sample_server.py execution screen (2. Get OTA Version Information)

```
>>> 3
 2020/06/23 11:49:07.921 Tx: D0EA83FC 0008 001F 0360 065D FE80000000000000021D12910000088A7CC
57CC500050101629D03
 2020/06/23 11:49:07.946 Rx: D0F9EE5D 2008 000B 0347 0105 01000101629D03
Get OTA Write BANK Information Request [SEND]:Success
 2020/06/23 11:49:07.962 Rx: D0F9EE5D 6018 0026 03B2 07E9 FE80000000000000021D12910000088A7CC
57CC555550002DE000701037206018403
 OTA Update Packet: 01 03 72 0601 84 03
Get OTA Write BANK Information Response [RECV:Success]
 OTA Write BANK = BANK1
```

Fig. 9: sample_server.py execution screen (3. Get OTA Write BANK Information)

```
>>> 4
Please input file path
        ex: bank0.hex
>>> BP35C0_J11_040001030000158A_OTA_Bank1.hex
loading hex file...
set FW Data 'BP35C0_J11_040001030000158A_OTA_Bank1.hex'
 min:14000A00,max:1403DFFF,NumOfSecotr:438
 2020/06/23 11:51:48.754 Tx: D0EA83FC 0008 0027 0368 0865 FE80000000000000021D12910000088A7CC
57CC5000D01094014000A001403DFFFA403
 2020/06/23 11:51:48.790 Rx: D0F9EE5D 2008 000B 0347 005F 01000109401400
Start OTA Write Request [SEND]:Success
 2020/06/23 11:51:48.809 Rx: D0F9EE5D 6018 0025 03B1 07E8 FE80000000000000021D12910000088A7CC
57CC555550002DE0006010270068803
 OTA Update Packet: 01 02 70 06 88 03
Start OTA Write Response [RECV:Success]
```

Fig. 10: sample_server.py execution screen (4. Start OTA Write)

```
>>> 5
OTA Data Write 0.23%(1/438)
 2020/06/23 11:52:59.253 Tx: D0EA83FC 0008 0221 0364 9976 FE80000000000000021D12910000088A7CC
57CC50207020001020078FC0020D50A0014E10A0014E30A001400000000000000000000000000000000000000000
0000000000000E50A0014000000000000000000E70A0014E90A0014AF5F0014EB0A0014C1170014ED5E00144D18001
4A95A0014F75E0014015F0014EB0A00144F1800147D5F0014EB0A0014EB0A0014EB0A0014875F0014915F00140B5F
001409170014155F0014EB0A0014EB0A0014EB0A0014EB0A0014EB0A0014EB0A00142D170014D75E0014EB0A0014E
B0A0014A15A0014EB0A0014EB0A00140348854600F0DCFC00480047754801147BFC00200548854605488047054800
47FEE7FEE7FEE7FEE7FEE7FEE700000120DD5E0014C10A001430B50B4601460020202022012409E00D46D5409D4205D
31D469540491B254695404019154652 1E002DF1DC30BD70B500242546002801DA01244042002901DA01254942FFF7
DDFFAC4200D04042002C00D0494270BD0549064A08685043054A8018086040087047014908607047881200206D4EC
6413930000010B5431A934209D28318881803E0401E01785B1E1970521EF9D210BD03460B439B0703D009E008C912
1F08C0042AFAD203E00B780370401C491C521EF9D210BDD2B201E00270401C491EFBD270470022F6E710B513460A4
604461946FFF7F0FF204610BDCAB2401E4178401C8A4202D00029F9D100207047421C0178401C0029FBD1801A7047
30B504460020034600E0AA17
 2020/06/23 11:52:59.396 Rx: D0F9EE5D 2008 000B 0347 0006 01000200010200
OTA Data Write Request [SEND(DataNum=1)]:Success
 2020/06/23 11:52:59.425 Rx: D0F9EE5D 6018 002C 03B8 09F0 FE80000000000000021D12910000088A7CC
57CC555550002DE000D0200010006061D1377FD97B803
 OTA Update Packet: 02 0001 0006 061D1377FD97 B8 03
OTA Data Write Response [RECV(DataNum=1):Success]
```

Fig. 11: sample_server.py execution screen (5. OTA Data Write)

```
>>> 6
 2020/06/23 11:57:29.314 Tx: D0EA83FC 0008 001F 0360 065D FE80000000000000021D12910000088A7CC
57CC50005010145BA03
 2020/06/23 11:57:29.345 Rx: D0F9EE5D 2008 000B 0347 0105 0100010145BA03
End OTA Write Request [SEND]:Success
 2020/06/23 11:57:29.506 Rx: D0F9EE5D 6018 0025 03B1 07E8 FE80000000000000021D12910000088A7CC
57CC555550002DE0006010275068303
 OTA Update Packet: 01 02 75 06 83 03
End OTA Write Response [RECV:Success]
```

Fig. 12: sample_server.py execution screen (6. End OTA Write)

```
>>> 7
 2020/06/23 11:57:31.138 Tx: D0EA83FC 0008 001F 0360 065D FE80000000000000021D12910000088A7CC
57CC500050101649B03
 2020/06/23 11:57:31.171 Rx: D0F9EE5D 2008 000B 0347 0105 01000101649B03
End OTA Mode Request [SEND]:Success
 2020/06/23 11:57:31.314 Rx: D0F9EE5D 6018 0025 03B1 07E8 FE80000000000000021D12910000088A7CC
57CC555550002DE0006010274068403
 OTA Update Packet: 01 02 74 06 84 03
End OTA Mode Response [RECV:Success]
```

Fig. 13: sample_server.py execution screen (7. End OTA Mode)

3.4   Checking update result on OTA Client

Check the version upgrade result of Notify OTA Operation Termination on OTA Client.

When "Version upgrade succeeded" appears, OTA update succeeded.

To reflect updated firmware after OTA update, execute Reset Hardware on the Wi-SUN module.


```
Waiting for OTA Update Completion...
 2020/06/23 11:49:04.640 Rx: D0F9EE5D 6033 0014 03BB 0251 FE80000000000000021D12910000060B
Notify OTA Operation Initiation: OTA Server=FE80000000000000021D12910000060B
 2020/06/23 11:57:31.170 Rx: D0F9EE5D 6034 0015 03BD 0252 01FE80000000000000021D12910000060B
Notify OTA Operation Termination: result=01 OTA Server=FE80000000000000021D12910000060B
 Version upgrade succeeded
 2020/06/23 11:57:31.271 Tx: D0EA83FC 0202 0004 0341 0000
 2020/06/23 11:57:31.312 Rx: D0F9EE5D 2202 0005 033D 0001 01
Terminate OTA Client:Success
```

Fig. 14: sample_client.py execution screen (OTA update success)

4.   File description

Table 5: Files list

File name	Description
`sample_server.py`	OTA Server main module
`sample_client.py`	OTA Client main module
`isb_wsn_ota.py`	Wi-SUN stack OTA update communication control module
`isb_wsn_serial.py`	Wi-SUN stack serial communication control module

4.1   Command option of sample_server.py

Table 6: Command option of sample_server.py

Option	Description
`-s`	Serial port device file Example: /dev/ttyUSB0
`-c`	Channel number (decimal) Example: 14
`-p`	PAN ID 2 bytes (hexadecimal character strings) Example: 5555
`-ca`	MAC address of OTA Client 8 bytes (hexadecimal character strings) Example: 001D12910000088A
`-cp`	Password of HAN PANA of OTA Client 16 bytes (hexadecimal character strings) Example: 30313233343536373839414243444546

4.2   Command option of sample_client.py

Table 7: Command option of sample_client.py

Option	Description
`-s`	Serial port device file Example: /dev/ttyUSB1
`-c`	Channel number (decimal) Example: 14
`-sa`	MAC address of OTA Server 8 bytes (hexadecimal character strings) Example: 001D12910000060B
`-cp`	Password of HAN PANA of OTA Client 16 bytes (hexadecimal character strings) Example: 30313233343536373839414243444546