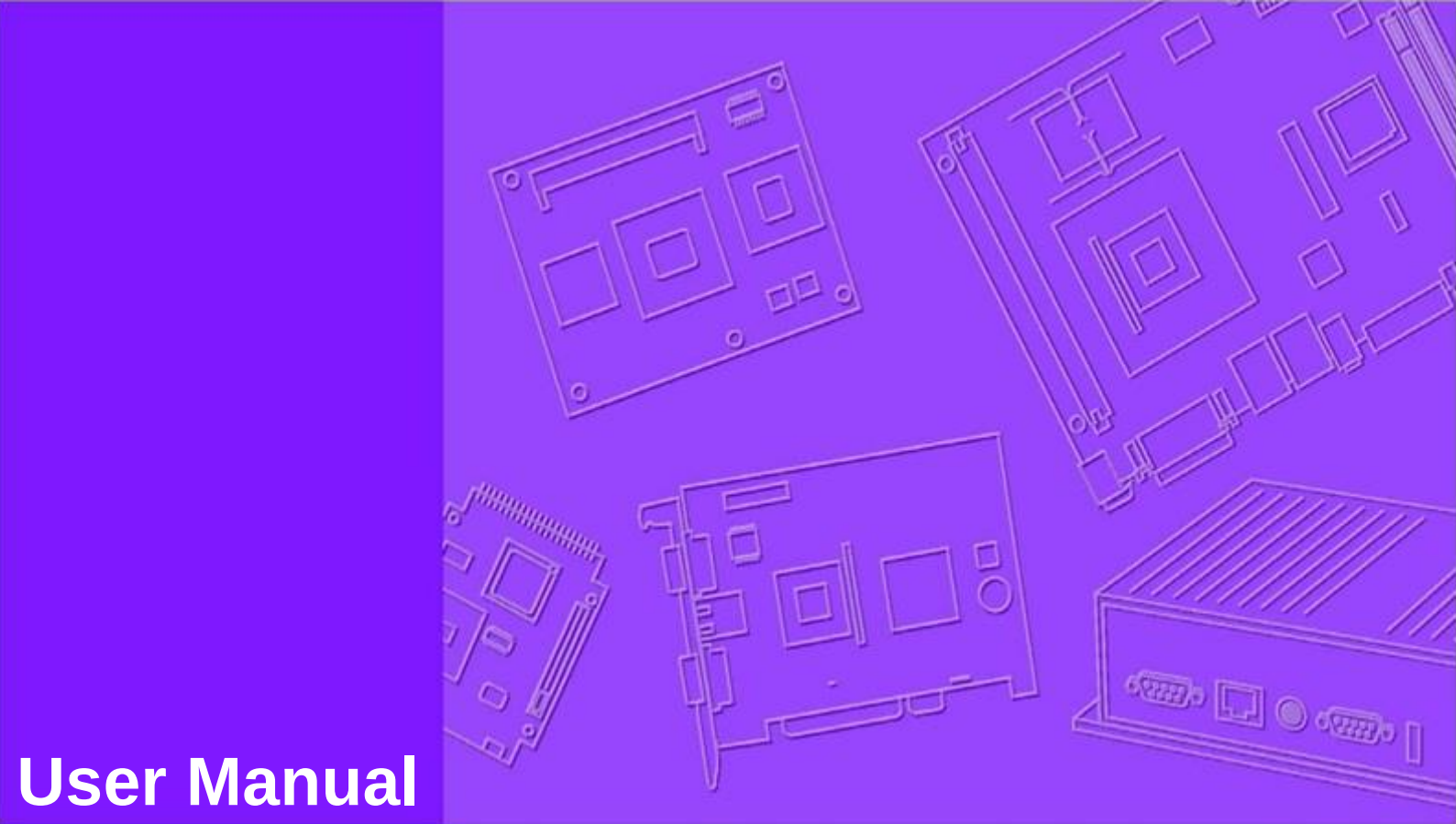




User Manual

WISE-1540

M2.COM Mesh Network IoT Node

ADVANTECH

Enabling an Intelligent Planet

Copyright

The documentation and the software included with this product are copyrighted 2016 by Advantech Co., Ltd. All rights are reserved. Advantech Co., Ltd. reserves the right to make improvements in the products described in this manual at any time without notice. No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission of Advantech Co., Ltd. Information provided in this manual is intended to be accurate and reliable. However, Advantech Co., Ltd. assumes no responsibility for its use, nor for any infringements of the rights of third parties, which may result from its use.

Acknowledgements

ARM is trademarks of ARM Corporation.

Freescale is trademarks of Freescale Corporation.

Microsoft Windows are registered trademarks of Microsoft Corp.

All other product names or trademarks are properties of their respective owners.

Product Warranty (2 years)

Advantech warrants to you, the original purchaser, that each of its products will be free from defects in materials and workmanship for two years from the date of purchase. This warranty does not apply to any products which have been repaired or altered by persons other than repair personnel authorized by Advantech, or which have been subject to misuse, abuse, accident or improper installation. Advantech assumes no liability under the terms of this warranty as a consequence of such events.

Because of Advantech's high quality-control standards and rigorous testing, most of our customers never need to use our repair service. If an Advantech product is defective, it will be repaired or replaced at no charge during the warranty period. For out-of-warranty repairs, you will be billed according to the cost of replacement materials, service time and freight. Please consult your dealer for more details.

If you think you have a defective product, follow these steps:

1. Collect all the information about the problem encountered. (For example, CPU speed, Advantech products used, other hardware and software used, etc.) Note anything abnormal and list any onscreen messages you get when the problem occurs.
2. Call your dealer and describe the problem. Please have your manual, product, and any helpful information readily available.
3. If your product is diagnosed as defective, obtain an RMA (return merchandise authorization) number from your dealer. This allows us to process your return more quickly.
4. Carefully pack the defective product, a fully-completed Repair and Replacement Order Card and a photocopy proof of purchase date (such as your sales receipt) in a

shippable container. A product returned without proof of the purchase date is not eligible for warranty service.

5. Write the RMA number visibly on the outside of the package and ship it prepaid to your dealer.

Declaration of Conformity

FCC Class B

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation.

This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

FCC Caution: Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

IMPORTANT NOTE:

FCC Radiation Exposure Statement:

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

This module is intended for OEM integrator. The OEM integrator is responsible for the compliance to all the rules that apply to the product into which this certified RF module is integrated. Additional testing and certification may be necessary when multiple modules are used.

USERS MANUAL OF THE END PRODUCT:

In the users manual of the end product, the end user has to be informed to keep at least 20cm separation with the antenna while this end product is installed and operated. The end user has to be informed that the FCC radio-frequency exposure guidelines for an uncontrolled environment can be satisfied.

The end user has to also be informed that any changes or modifications not expressly approved by the manufacturer could void the user's authority to operate this equipment.

If the labelling area is small than the palm of the hand, then additional FCC part 15.19 statement is required to be available in the users manual: This device complies with Part 15 of FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference and (2) this device must accept any interference received, including interference that may cause undesired operation.

LABEL OF THE END PRODUCT:

The final end product must be labeled in a visible area with the following " Contains TX FCC ID: M82-WISE1540 ".

If the labelling area is larger than the palm of the hand, then the following FCC part 15.19 statement has to also be available on the label: This device complies with Part 15 of FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference and (2) this device must accept any interference received, including interference that may cause undesired operation.

IC

This radio transmitter has been approved by Industry Canada to operate with the antenna types listed below with the maximum permissible gain indicated. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device.

Le présent émetteur radio a été approuvé par Industrie Canada pour fonctionner avec les types d'antenne énumérés ci-dessous et ayant un gain admissible maximal d'antenne. Les types d'antenne non inclus dans cette liste, ou dont le gain est supérieur au gain maximal indiqué, sont strictement interdits pour l'exploitation de l'émetteur.

This device complies with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes : (1) l'appareil ne doit pas produire de

brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

This radio transmitter (9404A-WISE1540) has been approved by Industry Canada to operate with the antenna types listed below with the maximum permissible gain indicated. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device.

Le présent émetteur radio (9404A-WISE1540) a été approuvé par Industrie Canada pour fonctionner avec les types d'antenne énumérés ci-dessous et ayant un gain admissible maximal d'antenne. Les types d'antenne non inclus dans cette liste, ou dont le gain est supérieur au gain maximal indiqué, sont strictement interdits pour l'exploitation de l'émetteur.

Antenna list:

Part No.	MPN	Description
1750008001-01	AN2450-92K01BRS	Antenna 2.4G/5GHz 5dBi SMA Male Reverse

IMPORTANT NOTE:

IC Radiation Exposure Statement:

This equipment complies with IC RSS-102 radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

Cet équipement est conforme aux limites d'exposition aux rayonnements IC établies pour un environnement non contrôlé. Cet équipement doit être installé et utilisé avec un minimum de 20 cm de distance entre la source de rayonnement et votre corps.

This module is intended for OEM integrator. The OEM integrator is responsible for the compliance to all the rules that apply to the product into which this certified RF module is integrated.

Additional testing and certification may be necessary when multiple modules are used.

Any changes or modifications not expressly approved by the manufacturer could void the user's authority to operate this equipment.

USERS MANUAL OF THE END PRODUCT:

In the users manual of the end product, the end user has to be informed to keep at least 20cm separation with the antenna while this end product is installed and operated. The end user has to be informed that the IC radio-frequency exposure guidelines for an uncontrolled environment can be satisfied.

The end user has to also be informed that any changes or modifications not expressly approved by

the manufacturer could void the user's authority to operate this equipment. Operation is subject to the following two conditions: (1) this device may not cause harmful interference and (2) this device must accept any interference received, including interference that may cause undesired operation.

LABEL OF THE END PRODUCT:

The final end product must be labeled in a visible area with the following " Contains IC: 9404A-WISE1540 ".

The Host Model Number (HMN) must be indicated at any location on the exterior of the end product or product packaging or product literature which shall be available with the end product or online.

低功率電波輻射性電機管理辦法

第十二條 經型式認證合格之低功率射頻電機，非經許可，公司、商號或使用者均不得擅自變更頻率、加大功率或變更原設計之特性及功能。

第十四條 低功率射頻電機之使用不得影響飛航安全及干擾合法通信；經發現有干擾現象時，應立即停用，並改善至無干擾時方得繼續使用。前項合法通信，指依電信法規定作業之無線電通信。低功率射頻電機須忍受合法通信或工業、科學及醫療用電波輻射性電機設備之干擾。

模組認證：

1. 本模組於取得認證後將依規定於模組本體標示審驗合格標籤。
2. 系統廠商應於平台上標示「本產品內含射頻模組：XXXyyyLPDzzzz-x」字樣。

Packing List

Before setting up the system, check that the items listed below are included and in good condition. If any item does not accord with the table, please contact your dealer immediately.

- 1 WISE-1540
- 1 Screw for WISE-1540
- 1 China RoHs Notice

Optional Accessories

Part No.	Description
1750008001-01	Antenna 2.4G/5GHz 5dBi SMA Male Reverse
1750007622-01	Antenna 2.4Ghz 3.5dBi SMA Male Reverse
1750007965-01	Antenna Cable R/P SMA (M) to MHF4, 300mm
1700015038	FPC Cable 10P-0.5mm 7.9cm for DCU2.0
9696WED200E	ASS'Y WISE-ED20 A101-1 M2.COM Daughter
1931000590	Screw M2.5x5L F/S D=5.3 H=0.8 (1+) ST Ni
1700023619-01	A cable USB-A 4P(M)/micro USB 5P(M) 1m ADAM-T212
1700025876-01	M cable USB-A 4P(M)/Plug-in 2P-5.0 90CM
XRISC-ADP-10HW-AG	ADP A/D 100-240V 10W 5V WM
193A231540	POST F=M3*6L M=M3*6L D=5 d=2.88 B=5 H=15 Cu

Development Board

Part No.	Description
9696150000E	ASS'Y WISE-DB1500 A101-1 M2.COM CARRIER

Ordering Information

Part No.	WISE-1540WMB-SDA10
Description	WISE-1540 M2.COM Mesh network IoT Node
Part No.	WISE-3310-D100L1E
Description	WISE-3310 Mesh Network IoT Gateway, 100-Nodes
Part No.	WISE-3310-D200L1E
Description	WISE-3310 Mesh Network IoT Gateway, 200-Nodes

Safety Instructions

1. Read these safety instructions carefully.
2. Keep this User Manual for later reference.
3. Disconnect this equipment from any AC outlet before cleaning. Use a damp cloth. Do not use liquid or spray detergents for cleaning.
4. For plug-in equipment, the power outlet socket must be located near the equipment and must be easily accessible.
5. Keep this equipment away from humidity.
6. Put this equipment on a reliable surface during installation. Dropping it or letting it fall may cause damage.
7. The openings on the enclosure are for air convection. Protect the equipment from overheating. **DO NOT COVER THE OPENINGS.**
8. Make sure the voltage of the power source is correct before connecting the equipment to the power outlet.
9. Position the power cord so that people cannot step on it. Do not place anything over the power cord.
10. All cautions and warnings on the equipment should be noted.
11. If the equipment is not used for a long time, disconnect it from the power source to avoid damage by transient overvoltage.
12. Never pour any liquid into an opening. This may cause fire or electrical shock.
13. Never open the equipment. For safety reasons, the equipment should be opened only by qualified service personnel.
14. If one of the following situations arises, get the equipment checked by service personnel:
 - The power cord or plug is damaged.
 - Liquid has penetrated into the equipment.
 - The equipment has been exposed to moisture.
 - The equipment does not work well, or you cannot get it to work according to the user's manual.
 - The equipment has been dropped and damaged.
 - The equipment has obvious signs of breakage.

DISCLAIMER: This set of instructions is given according to IEC 704-1. Advantech disclaims all responsibility for the accuracy of any statements contained herein.

Change Log:

Date	Version	Description / Major change
2017/10/24	V0.1	Draft version
2017/10/26	V0.2	Update NCC Statement on page 5

Table of Content

1. PRODUCT OVERVIEW	11
1.1. INTRODUCTION	11
1.2. SPECIFICATIONS	12
2. H/W INSTALLATION	13
2.1. BOARD CONNECTOR	13
2.2. MODULE OUTLINE	15
2.3. CONNECTOR SPECIFICATIONS	15
2.3.1. TOP SIDE CONNECTOR PHYSICAL DIMENSIONS	15
2.3.2. CARRIER BOARD CONNECTION LENGTH	16
2.3.3. CARRIER BOARD CONNECTOR HEIGHT	17
2.4. WISE-1540 PIN-OUT MAP	18
3. DEVELOPMENT ENVIRONMENT SETUP	20
3.1. OVERVIEW	20
3.2. INSTALLATION	20
3.3. ARM MBED CONFIGURATION	21
3.4. SOURCE TREE	22
3.5. SETUP DEVELOPMENT ENVIRONMENT	22
3.6. SETTING FOR CONNECTIVITY	26
3.7. FLASH PROGRAMMING	29
<i>Programming through WISE-ED22</i>	<i>29</i>
<i>Programming through WISE-ED20</i>	<i>29</i>
4. EXAMPLES	36
4.1. COMPILE EXAMPLE	36
4.2. SEND DATA	36
<i>Running example on Gateway</i>	<i>36</i>
<i>Running example on WISE-1540</i>	<i>39</i>
4.3. EXCHANGE DATA WITH SENSOR FORMAT	40
REFERENCES	42

1. Product Overview

1.1.Introduction

WISE-1540 is M2.COM standardized IoT Sensor Node integrated with ARM Cortex-M4 Processor and Dust's time-synchronized SmartMesh networks. With time slotted channel hopping (TSCH) link layer, all nodes are synchronized to within less than a millisecond. Time in the network is organized into time slots, which enables collision-free packet exchange and per-transmission channel-hopping.

All nodes within SmartMesh networks could route, source or terminate data while providing years of battery powered operation. SmartMesh Networks are the great fit for Industry 4.0 Applications to deliver a highly reliable network in the most challenging RF environments such as Machine diagnostics, Factory Automation, Environment Monitoring and Production Efficiency. In the SmartMesh network, every device has one or more parents which provide redundant paths to overcome communications interruption due to interference, physical obstruction or multi-path fading. If a packet transmission fails on one path, the next retransmission may try on a different path and different RF channel. Developer can be easily to extend the network coverage via SmartMesh.

Advantech WISE-1540 also provides multi-interfaces for sensor and I/O control. With Arm Mbed embedded microprocessor operating system and add-on software stacks, it's convenient to support IoT communication protocols including LWM2M, CoAP and MQTT via WISE-3310 Mesh IoT Gateway. Data can be quickly and easily acquired and transformed into a different format to communicate with WISE-PaaS, Mbed Cloud or other cloud services. Developer can build their application backbone faster and focus on their applications, value-added services.

The main features of WISE-1540 are:

- ARM Cortex-M4 Core Processor
- IEEE 802.15.4e standard with self-healing mesh network
- 99.999% Data Reliability and power optimization
- Rich interface for sensor and I/O control
- Support wide temperature -40 ~ 85 °C

1.2. Specifications

Processor System	MCU	ARM Cortex-M4 Core Processor 80MHz STM – STM32L443RC
Memory	RAM	64KB
	Flash	256KB
Form Factor		M2.COM Type A 2230
Spec. Standard		M2 COM Technical SPEC_v1.1
Wireless Network	Standard	IEEE 802.15.4e
	Frequency Band	2.4000~2.4835 GHz
	Channels	15
	Topology	Self-Healing Mesh Network
	Transmit Power	Up to +8dBm
	Receiver Sensitivity	Up to -93dBm
	RF Data Rate	250 kbps
	Function	End node
	Antenna connector	MHF4 connector
I/O	UART	1 (4-wire, support RTC/CTS)
	I2C	1
	GPIO	8
	PWM	1
	SPI	1
	ADC	4
	USB	1 (device only)
Programming / Debug Port		1 via WISE-ED22 (CN1)
Power		3.3V
Environment	Operational Temperature	-40 ~ 85° C
	Operating Humidity	5% ~ 95% Relative Humidity, non-condensing
Physical Characteristics	Dimensions (WxD)	22 x 30 mm

2. H/W Installation

2.1.Board Connector

- M2.COM Type A Module
 - Module size: 22 mm x 30 mm
 - PCB thickness: 0.8 mm $\pm$ 10%
 - Pin count: 75 pins
 - Module input voltage: 3.3V DC-in
 - Connector mating force: 30N Maximum
 - Connector current rating: 0.5A / Power contact
 - Connector operation temperature range: -45 °C to +85 °C

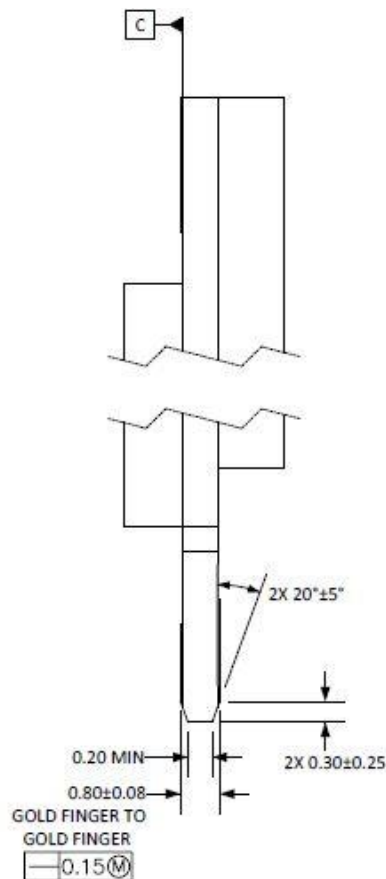


Figure 1 Card Edge Bevel

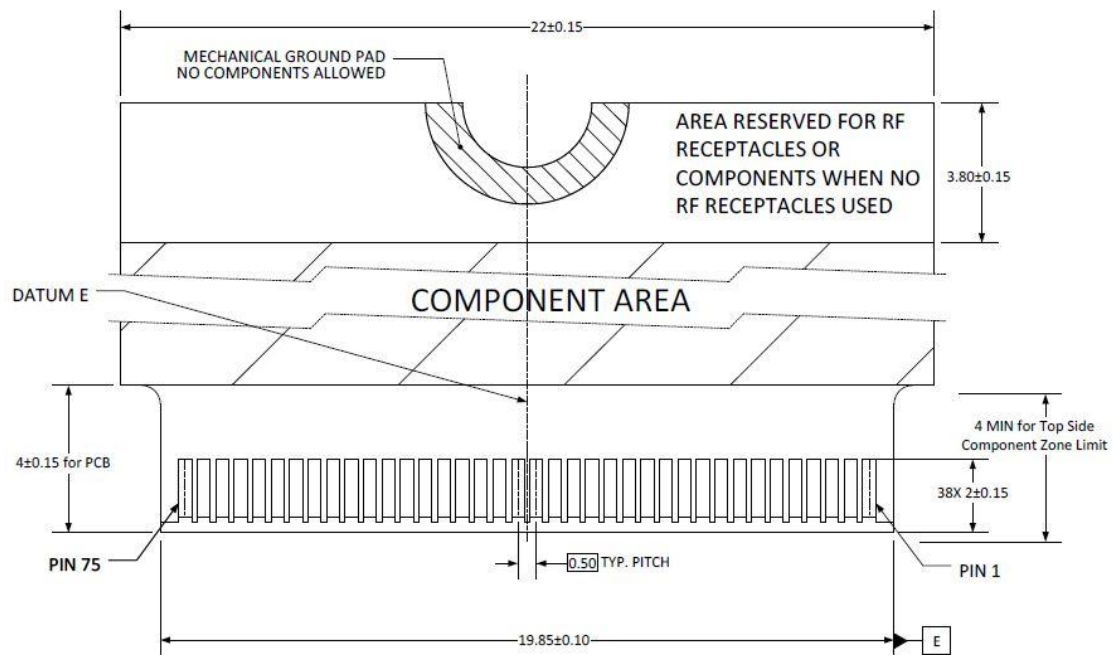


Figure 2 Card Edge Outline-Topside

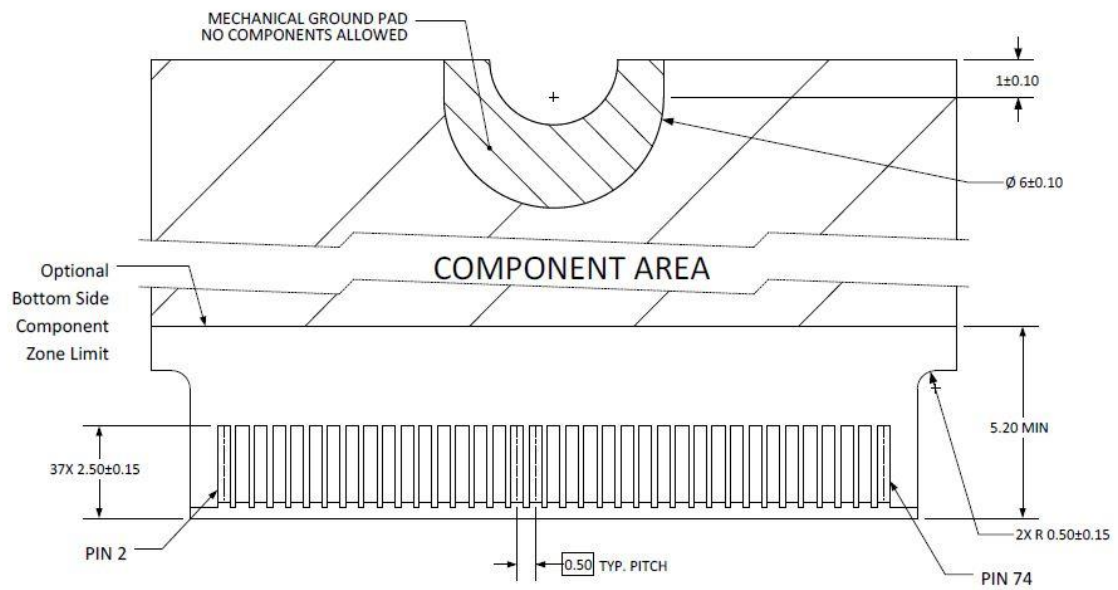


Figure 3 Card Edge Outline-Backside

Reference from PCI Express M.2 Specification Rev 1.0 (Nov 1, 2013) Section 2.3.5 Card PCB Details

2.2.Module Outline

The mechanical dimension information of M2.COM form factor follows the Type A 2230 module size: 22 x 30 mm. Both module types use a 75-position host interface connector and have room to support up to four RF connectors in the upper section.

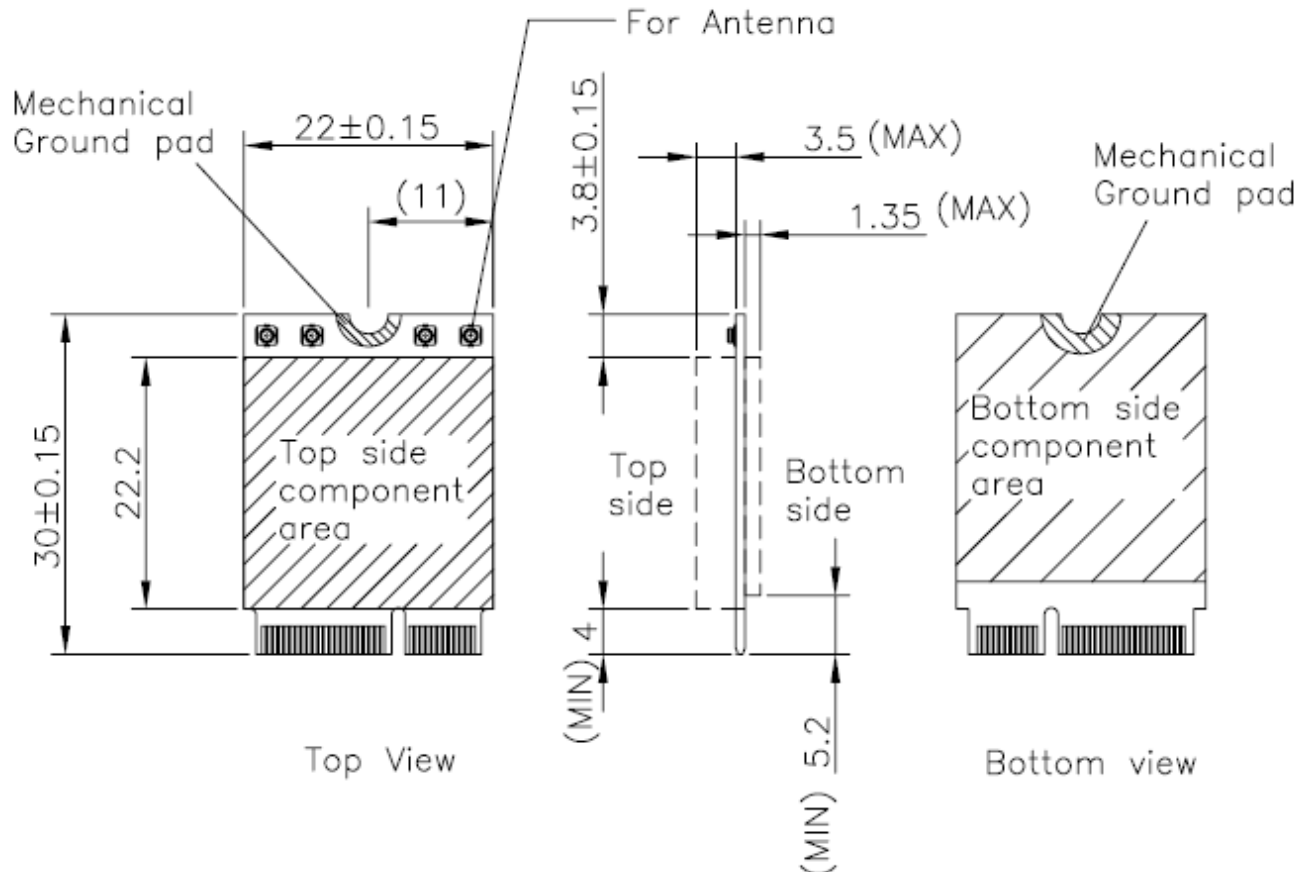


Figure 4 Type A 2230

2.3.Connector Specifications

2.3.1. Top Side Connector Physical Dimensions

The top-side scheme has two connectors that share a common footprint but have different stack-up requirements.

- Length – 22 mm maximum including land pattern
- Width – 9.1 mm maximum including land pattern

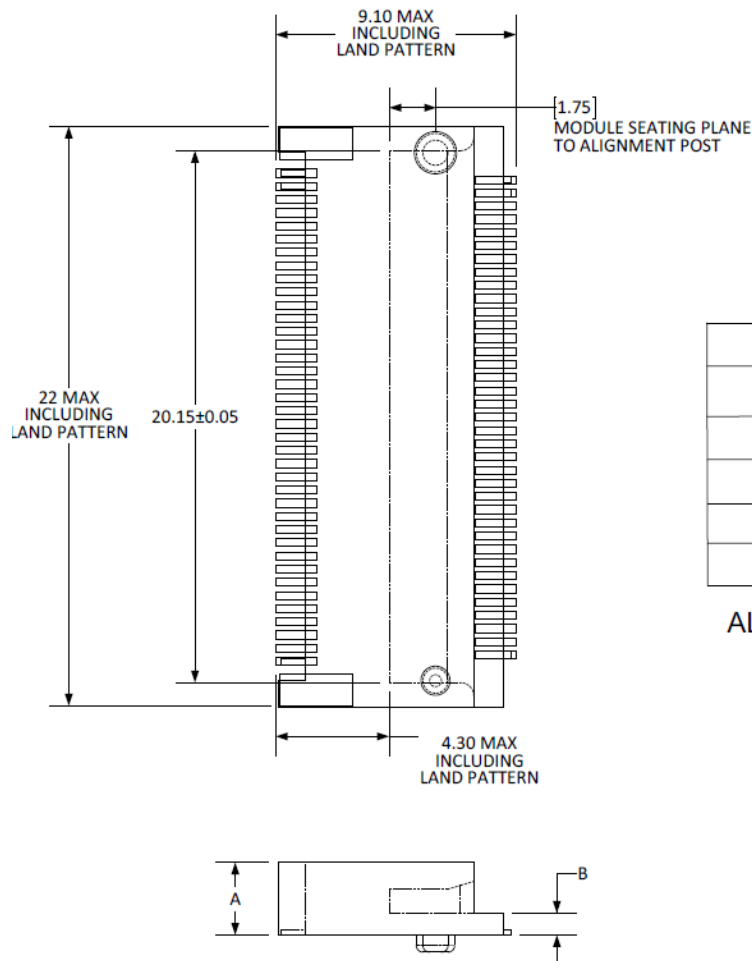


Figure 5 Top Side Connector Physical Dimensions

Reference from PCI Express M.2 Specification, Revision 1.0, November 1, 2013

2.3.2. Carrier Board Connection Length

The carrier board connector of M2.COM follows the Type 2230 M.2 module connector:

- The additional increase in length is 7.05mm maximum for top-side connector to the module length.
 - The retention screw adds 2.75 mm maximum.
 - The maximum extension, including land pattern, beyond the module leading edge is 4.3 mm.
- M2.COM module lengths are 30 mm and 42 mm.

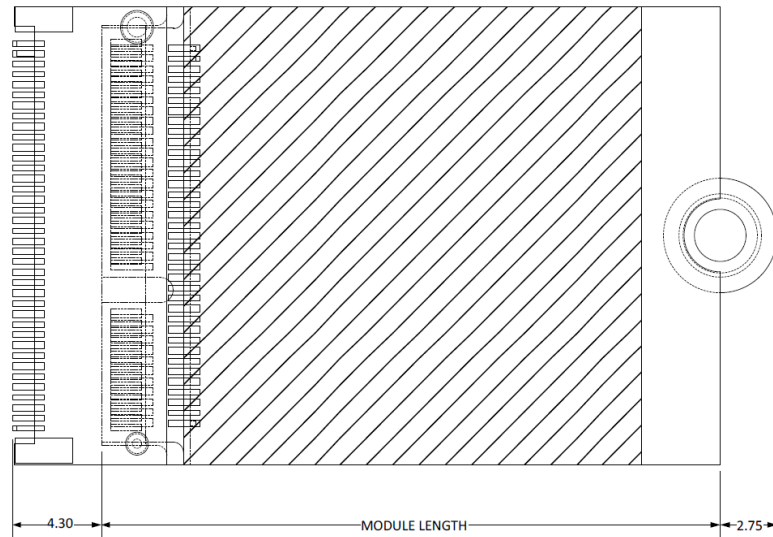


Figure 6 Carrier Board Connection Length

Reference from PCI Express M.2 Specification, Revision 1.0, November 1, 2013

2.3.3. Carrier Board Connector Height

The dimensions of M2.COM form factor follow the Type A 2230 -D3 M.2 module size. Hence, the carrier board connectors must choose H3.2-D3 or H4.2-D5 connector as in the following diagrams.

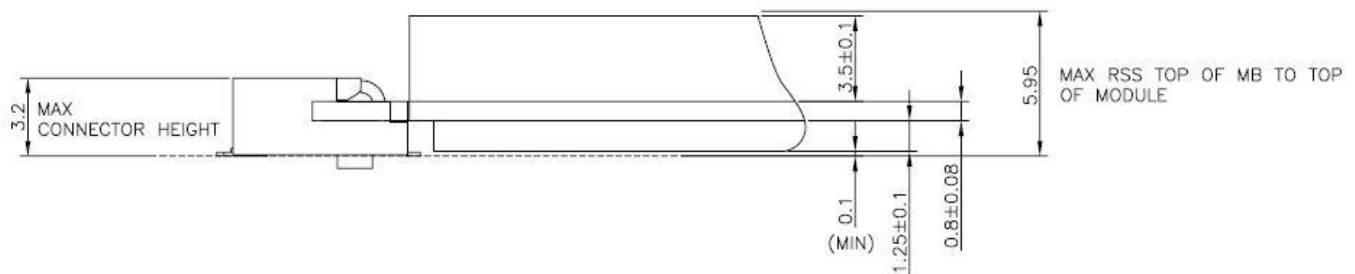


Figure 7 H3.2-D3

Reference from PCI Express M.2 Specification, Revision 1.0, November 1, 2013

2.4. WISE-1540 Pin-Out Map

PIN	M2.COM Signal name	STM32L443RCI6 MCU Pin Name		M2.COM Signal name	PIN
1	GND	GND	3.3V	VCC	2
3	USB_DP	PA12	3.3V	VCC	4
5	USB_DM	PA11		N.C.	6
7	GND	GND		N.C.	8
9	N.C.			N.C.	10
11	N.C.			N.C.	12
13	N.C.			N.C.	14
15	N.C.		PC6	CB_RESET_OUT#	16
17	N.C.		GND	GND	18
19	N.C.		PC9	CB_PWR_ON	20
21	N.C.		PA2	UART TX (O)(0/3.3V)	22
23	N.C.			Connector Key	
	Connector Key			Connector Key	
	Connector Key			Connector Key	
	Connector Key			Connector Key	
	Connector Key		PA3	UART RX (I)(0/3.3V)	32
33	GND	GND	PA1	UART RTS (O)(0/3.3V)	34
35	N.C.		PA0	UART CTS (I)(0/3.3V)	36
37	N.C.		PA8	GPIO0	38
39	GND	GND	PC8	GPIO1	40
41	PWM0	PA5	PC7	GPIO2	42
43	N.C.		PB13	GPIO3	44
45	GND	GND	PB0	GPIO4	46
47	ADC0	PA7	PB2	GPIO5	48
49	N.C.		PB1	GPIO6	50
51	GND	GND	PB6	GPIO7	52
53	ADC2	PA6		N.C.	54
55	ADC3	PA4	PC2	W_DISABLE#	56
57	GND	GND	PC1	I2C_DATA	58
59	ADC4	PC5	PC0	I2C_CLK	60
61	N.C.		PB15	SPI_MOSI	62
63	GND	GND	PB14	SPI_MISO	64
65	VDD_RTC	VBAT(3.3V)	PB10	SPI_CLK	66

67	Backup#	PB5	PB12	SPI_CS0#	68
69	GND	GND	PB9	SPI_CS1#	70
71	RESET_IN#	NRST	3.3V	VCC	72
73	Wake#	PC3	3.3V	VCC	74
75	GND	GND			

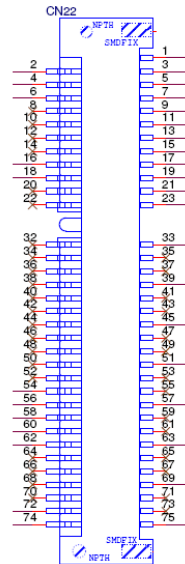


Figure 8 M.2 Connector

3. Development Environment Setup

3.1. Overview

ARM mbed is used for you to create applications running on WISE-1540. Your application code is written in C++. It uses the application programming interfaces (APIs) that mbed OS provides. These APIs allow your code to work on different microcontrollers in a uniform way. This reduces a lot of the challenges in getting started with microcontrollers and integrating large amounts of software. Besides, we also provide you node APIs which facilitates SmartMesh node development. Our offline development tool is the mbed CLI, a command-line tool. This requires having a toolchain installed on your computer. mbed CLI is the name of the ARM mbed command-line tool, packaged as mbed-cli, which enables the full mbed workflow: repositories version control, maintaining dependencies, publishing code, updating from remotely hosted repositories and invoking ARM mbed's own build system and export functions, among other operations. The basic workflow for mbed CLI is to:

1. Initialize a new repository, for either a new application (or library) or an imported one.
2. Build the application code.
3. Test your build.
4. Publish your application.

3.2. Installation

To install mbed CLI, related tools are required to be installed first. Please refer to the video tutorial. (<https://www.youtube.com/watch?v=cM0dFoTuU14>)

Please follow the steps described in the tutorial video to install mbed CLI.

1. Install Python

mbed CLI supports Windows, Linux and Mac OS X operating systems. You can select the OS you prefer to work with. mbed CLI is a Python script, so you'll need Python to use it. The version 2.7.11 of Python has been verified with mbed CLI.

<https://www.python.org/downloads/release/python-2711/>

Note: mbed CLI is incompatible with Python 3.

2. (Optional) Install Git or Mercurial

If you would like to maintain your source code in repositories, you can continue with the next step. mbed CLI supports both Git and Mercurial repositories, you can

install which one you prefer:

Git - version 1.9.5 or later (<https://git-scm.com/>).

Mercurial - version 2.2.2 or later (<https://www.mercurial-scm.org/>).

If you don't want to use repositories, you can just skip it.

3. Install gcc

mbed CLI invokes the mbed OS 5 tools for various features, such as compiling, testing and exporting to industry standard toolchains. To compile your code, you will need either a compiler or an IDE:

- Compilers: GCC ARM, ARM Compiler 5, IAR.
- IDE: Keil uVision, DS-5, IAR Workbench.

We select GCC ARM Embedded, so you can install version 4.9 of GCC ARM Embedded (<https://launchpad.net/gcc-arm-embedded>).

Note: Version 5.0 or any other versions above may be incompatible with the tools.

4. Install mbed CLI

You can get the latest stable version of mbed CLI from PyPI

```
$ pip install mbed-cli
```

Note: On Linux or Mac, you may need to run with sudo.

Finally, you've to extract the source code to the working directory from the SDK we released. The structure of the working directory is as below:

docs/	<-- Documents for SDK
loranode_L443_sdk_R1_0_02/mbed-os/	<-- mbed os
loranode_L443_sdk_R1_0_02/libHLLoraNode.a	<-- Harmony Link Lora
Node library	
loranode_L443_sdk_R1_0_02/node_api.h	<-- Node API header file
loranode_L443_sdk_R1_0_02/main.cpp	<-- Sample code

3.3.ARM mbed Configuration

The ARM mbed is IoT device platform and it has a lot of resources for IoT development. We supported ARM mbed OS on WISE-1540 to make user easily to get started and obtain great benefit from ARM mbed.

mbed CLI

The mbed CLI is command-line tool. The user needs to setup CLI for WISE-1540 SDK. Please refer to below link for information about how to setup CLI.

- mbed CLI

https://docs.mbed.com/docs/mbed-os-handbook/en/latest/dev_tools/cli/

mbed CLI configuration

- GCC_ARM_PATH

Set GCC_ARM_PATH to the binary directory of your GCC Arm installation

```
$ mbed config -G GCC_ARM_PATH
"/home/erick/gcc-arm-none-eabi-4_9-2015q3/bin"
[mbed] /home/erick/gcc-arm-none-eabi-4_9-2015q3/bin now set as global
GCC_ARM_PATH
```

3.4.Source Tree

The user can find source tree as below from WISE-1540-SDK.

(<https://github.com/ADVANTECH-Corp/WISE-1540-SDK>)

Directory	Description
doc/	Documents about WISE-1540 SDK.
inc/	Header files including examples, sensor driver and etc.
lib/	Library.
src/	Source files including examples, sensor driver and etc.
tool/	Tools.

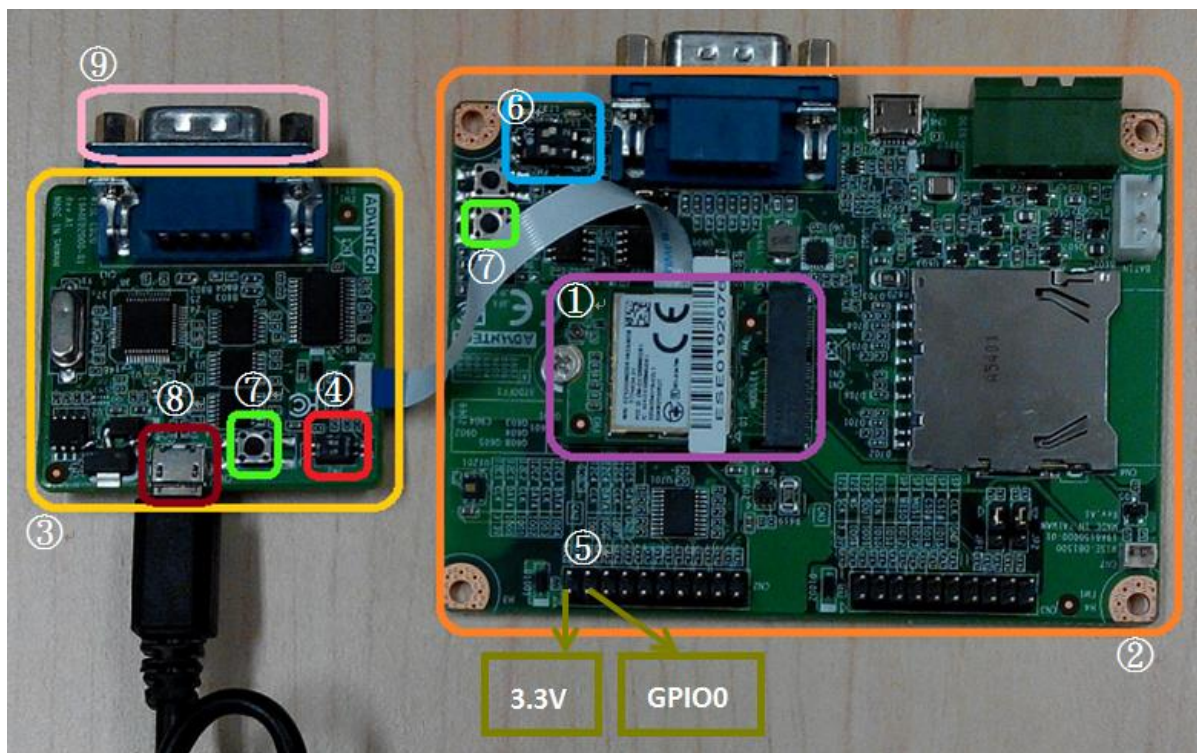
3.5.Setup Development Environment

The user needs to prepare for hardware as following:

- For mote, WISE-1540, WISE-1500 and WISE-ED20 or [WISE-ED22](#).
- For gateway, WISE-3310.
- PC running LINUX operating systems such as Ubuntu 16.04.1.

Please refer to the following steps for setup a WISE series of boards before using WISE-1540 SDK.

Step01: Please prepare boards as below.



① WISE-1540

The user must to choose the CN1 on WISE-1540 as below is connected to WISE-ED20 or WISE-ED22 through FPC.



② WISE-1500

③ WISE-ED20

④SW2 on WISE-ED20

- Please check pin1 and pin2 to “ON” as default.
- Switch pin1 to “ON”: flash programming, “OFF”: boot from flash.

⑤CN2 on WISE-1500

- Booting into the mode of network connection when both 3.3V and GPIO0 are opened.
- Booting into the mode of console of API when both 3.3V and GPIO0 are shorted.

⑥SW3 on WISE-1500

- Please check pin1 and pin2 to “OFF” as default.
- Switch pin2 to decide whether running “backup to default” while device booting.
“ON”: enabled, “OFF”: disabled.

⑦One reset button on WISE-1500 and the other on WISE-ED20

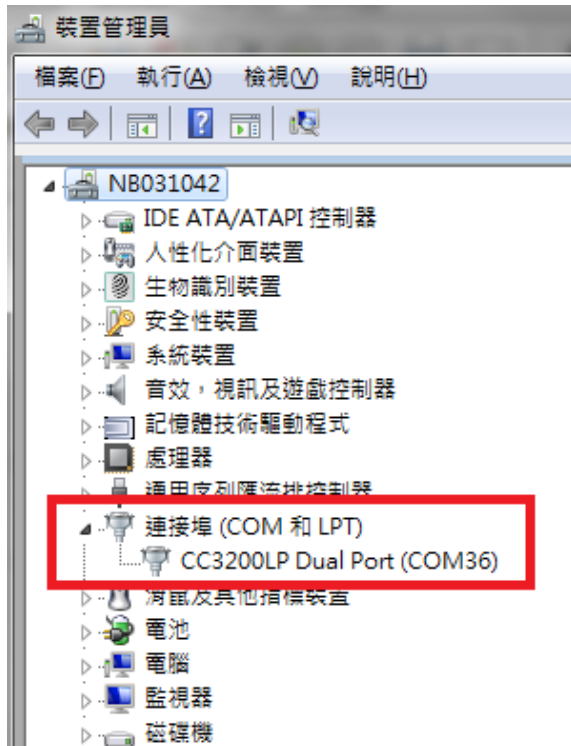
⑧Micro USB connector

- Power supplies for UART3 debug port.

⑨Com port

- No supported.

Step02: Connect the WISE-ED20 to PC using micro-USB cable.



The device will be visible in the Device Manager as below after FTDI driver installed.
The user can find the FTDI driver in source tree.

3.6.Setting for Connectivity

WISE-1540 is designed in connectivity for wireless IoT communication with Linear Dust SmartMesh IP wireless sensor networks. The Linear Dust networks support a self-forming and self-healing mesh network solution compliant to 6LoWPAN Internet Protocol (IP) and IEEE802.15.4e standard. Before using WISE-1540 SDK, the user needs to check or set some parameters through the console for API.

Parameter	Expected Setting
Network ID	2001 (The same as manager on WISE-3310)
Join Key	4A4F494E414456414E54454348494F54 (The same as manager on WISE-3310)
Auto Join	On

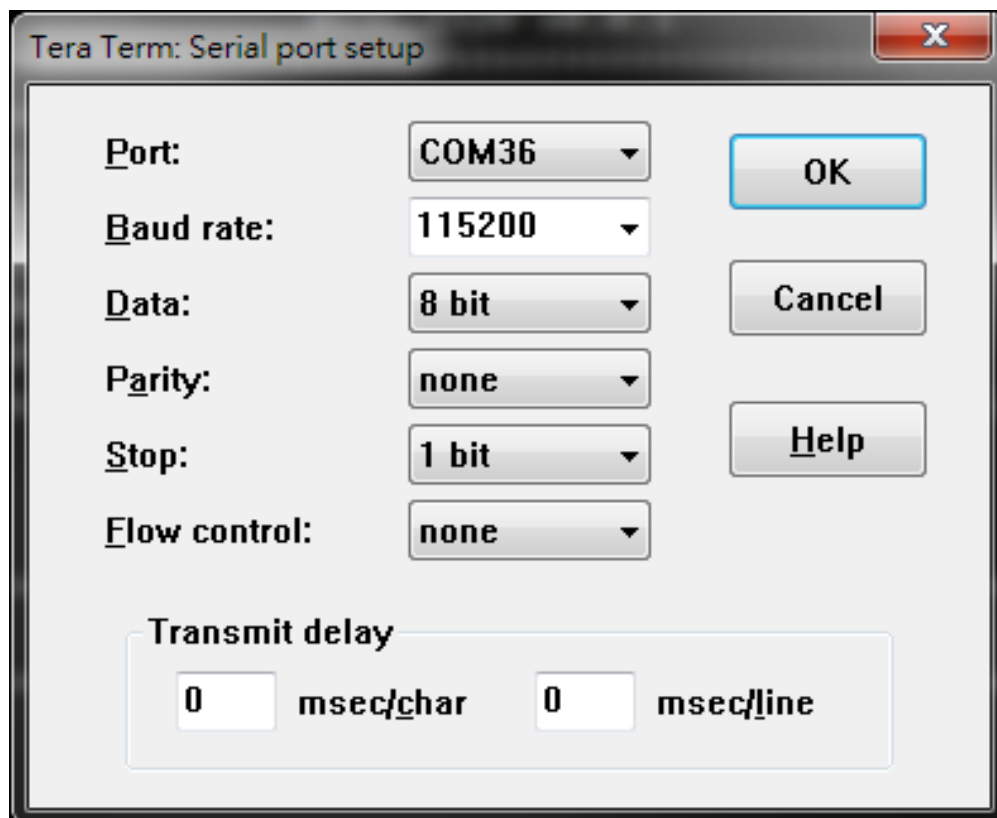
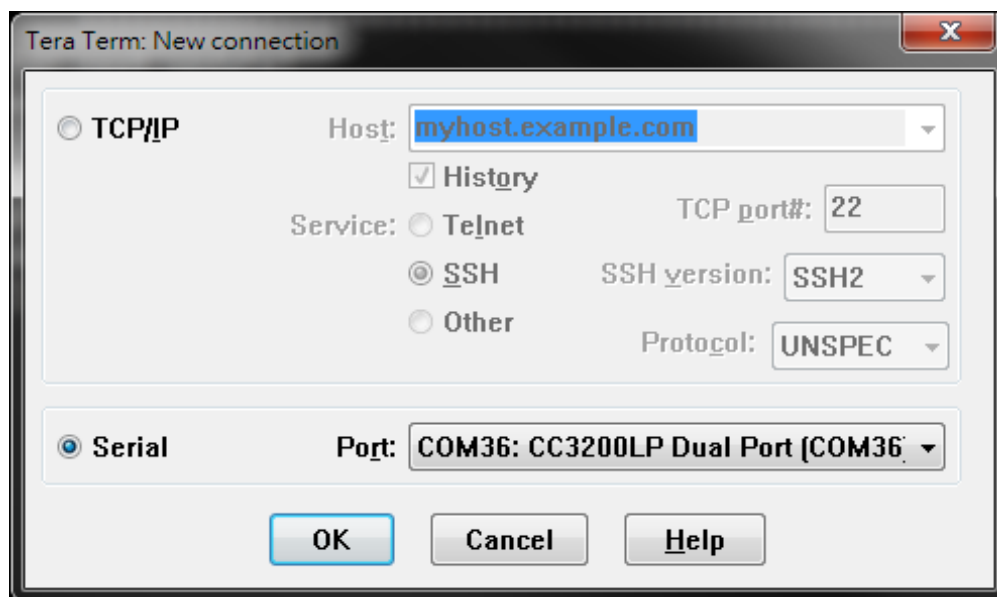
Reset to default setting

This function can be used back to default setting the same as above section "Information and Parameters Setting". The user can refer to section "Preparing for Hardware" for how to do it on the WISE-DB1500.

Console for API

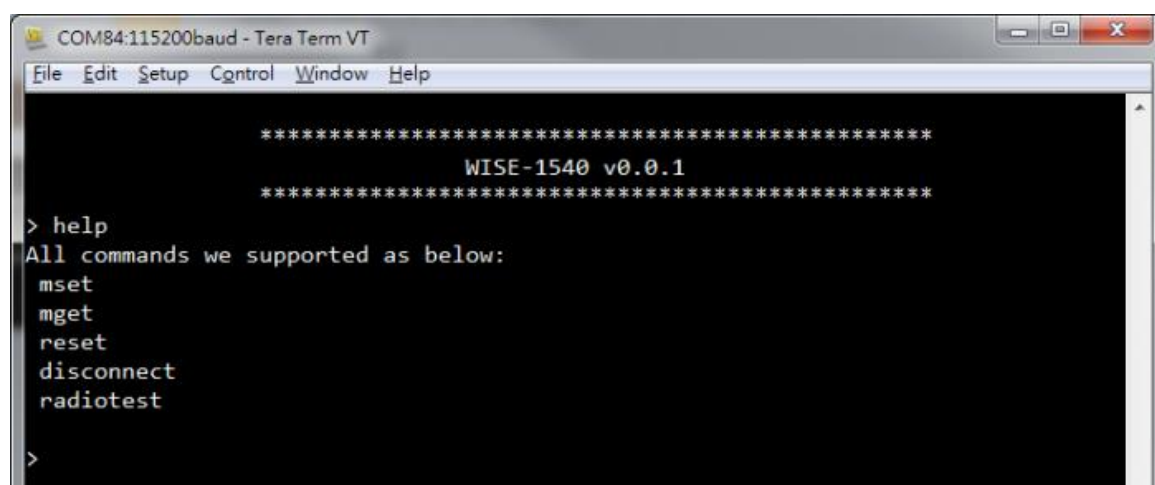
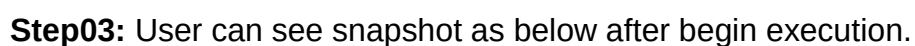
We provide the console and the user can set parameters for connectivity by command set we supported.

Step01: Check device connected from WISE-ED20 or WISE-ED22 to PC using micro-USB cable and choose any terminal you like. (e.g., the snapshot is setting of Tera Term.)



Step02: Check jumper.

Please check both 3.3V and GPIO0 of CN2 are shorted on WISE-1500 and press reset button on the WISE-ED20 or WISE-ED22 to restart device.



3.7.Flash Programming

For flash programming, the user can do it through [WISE-ED22](#) or WISE-ED20 with different programming tool. Please refer to the following steps for flash programming.

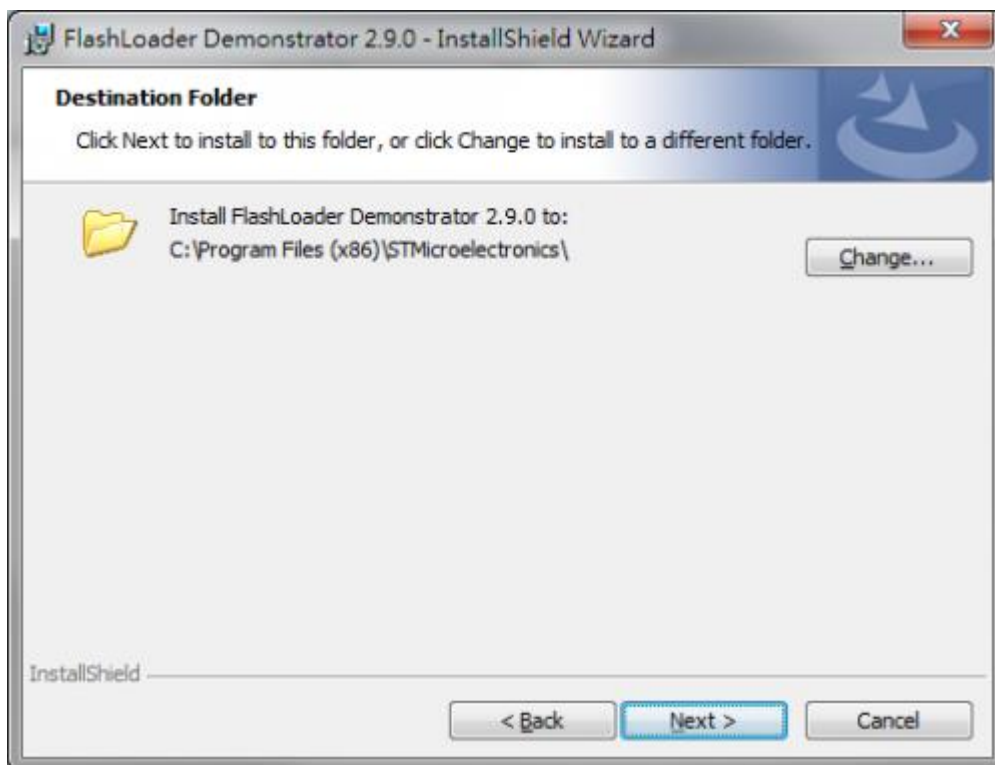
Programming through WISE-ED22

Please refer to [WISE-ED22](#) for flash programming.

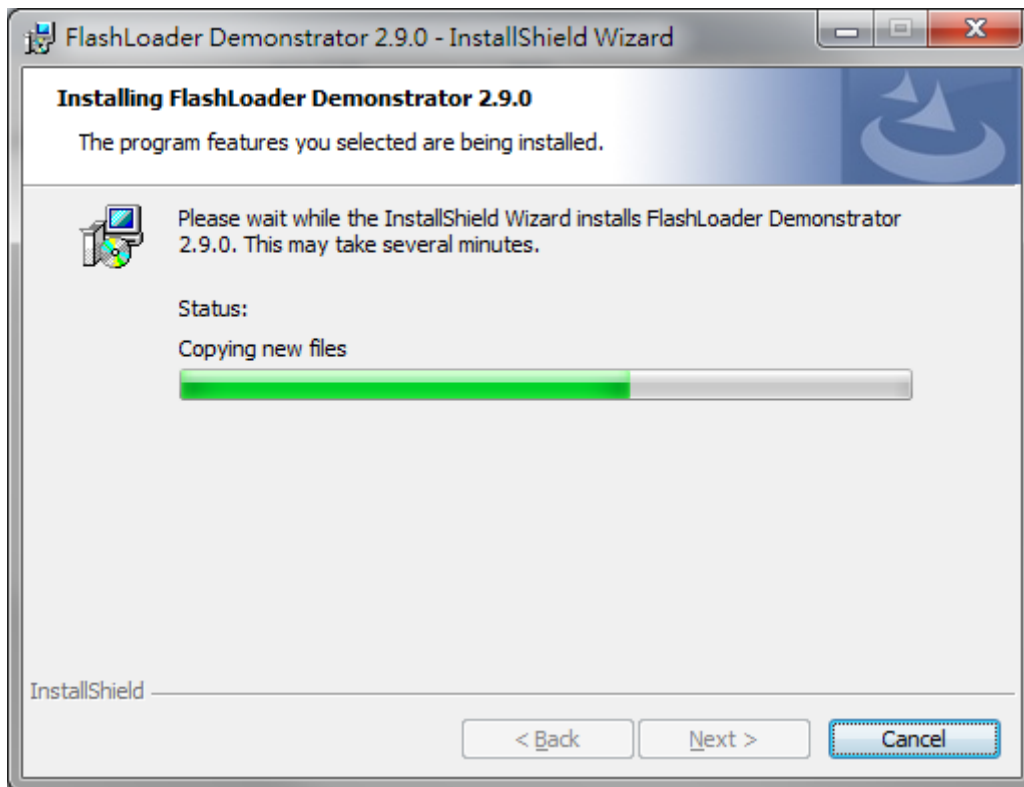
Programming through WISE-ED20

Running Flash Loader Installer

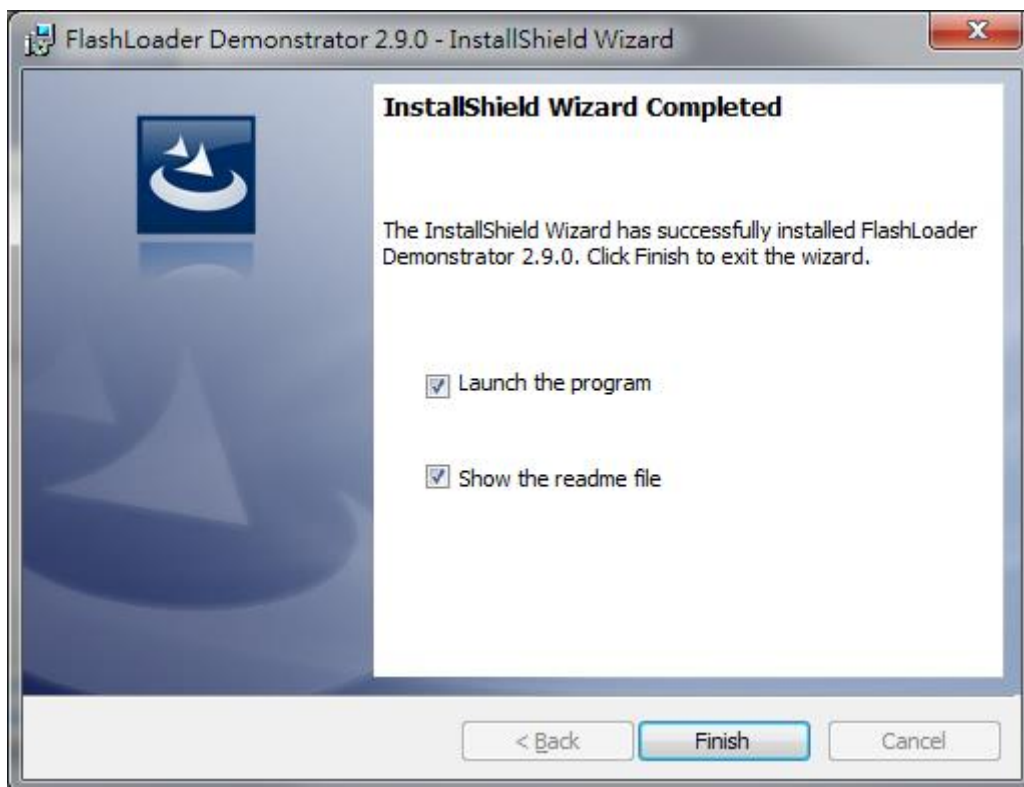
Step01: Find installer flash_loader_demo_v2.9.0RC1.exe” in source tree, run it to install tool and set installation directory to default as below.



Step02: Installation process.



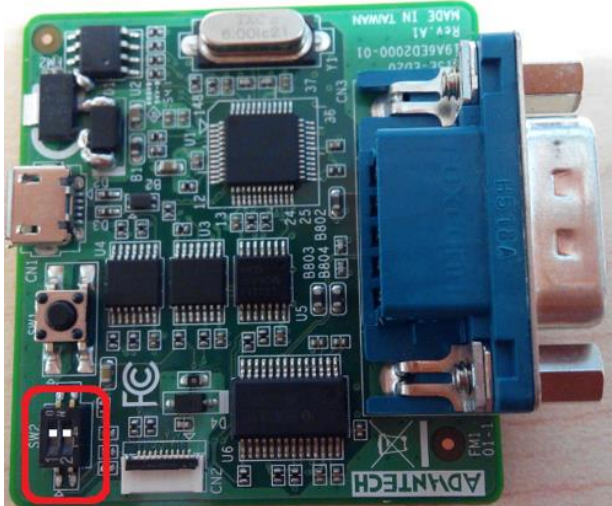
Step03: Finish installation.



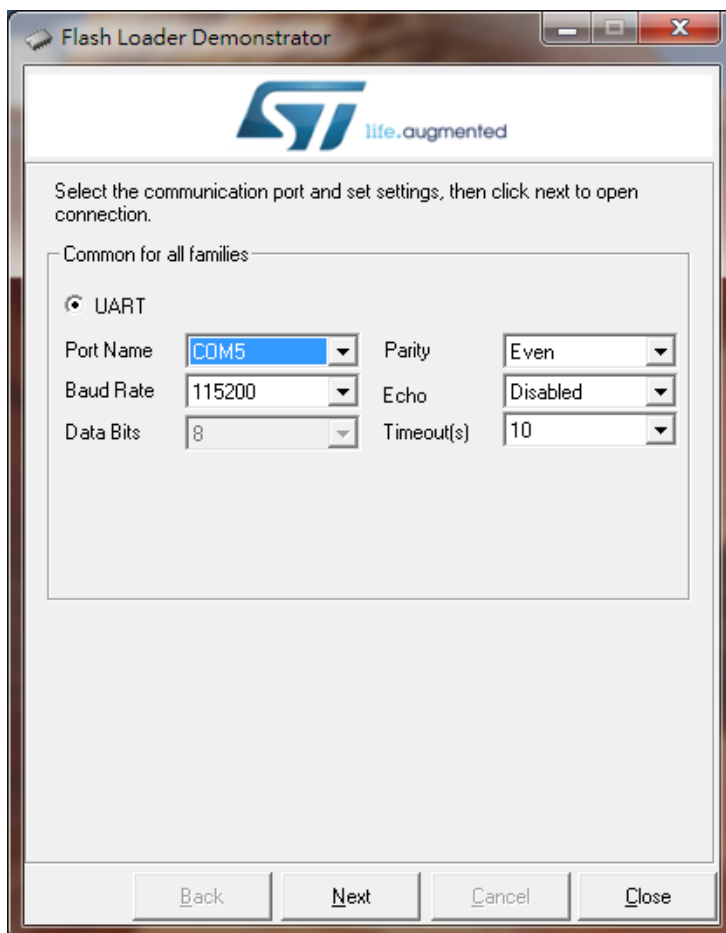
Programming

Step01 Set into programming mode.

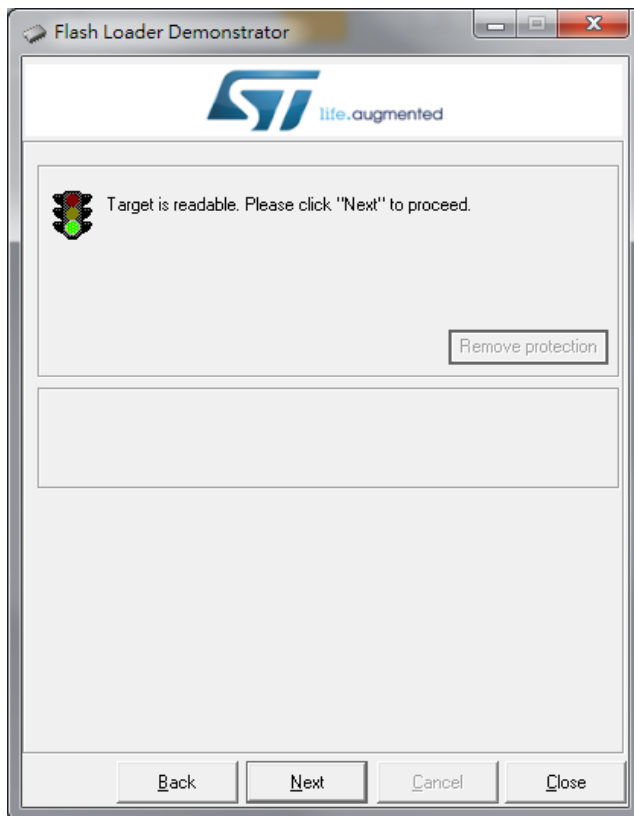
Please check pin1 of SW2 to “ON” on WISE-ED20 for flash programming.



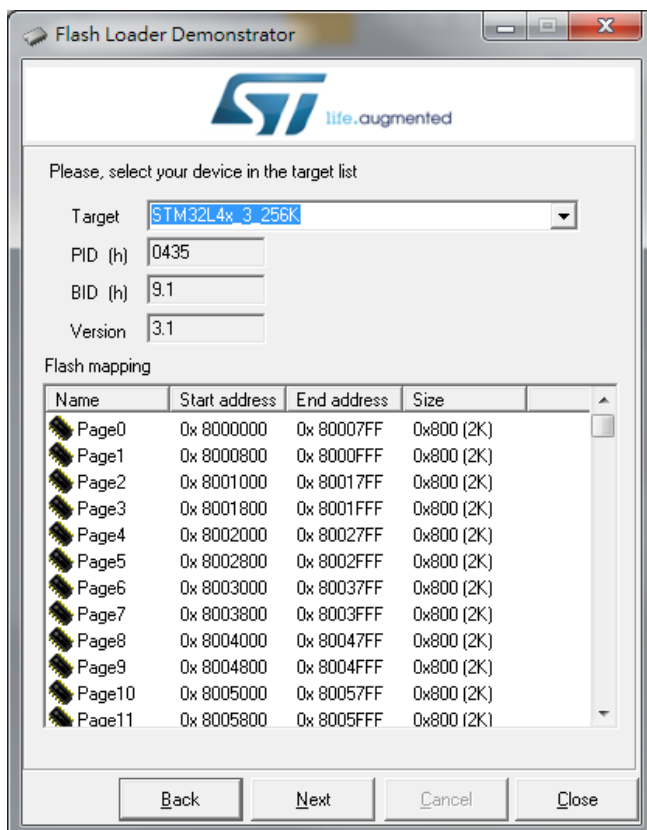
Step02: Launch Flash Loader and select Port name "COMx" detected from your PC.



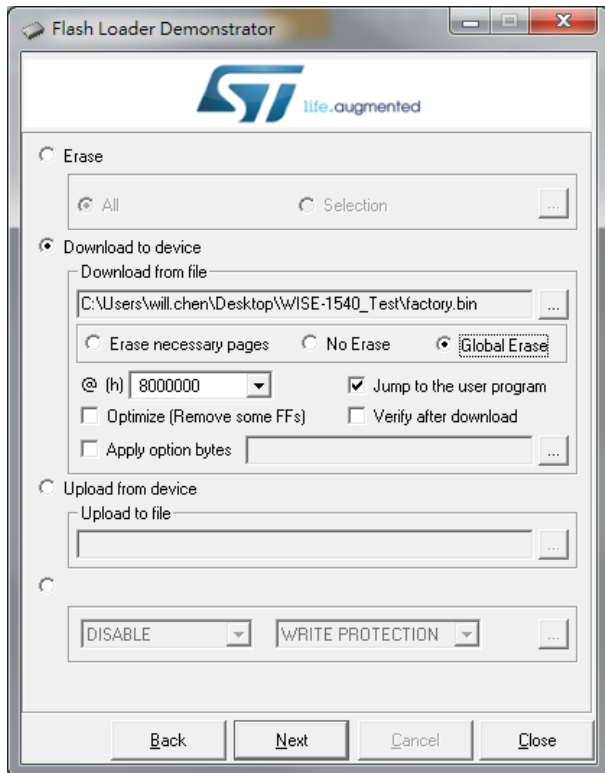
Step03: Check device if get response from it.



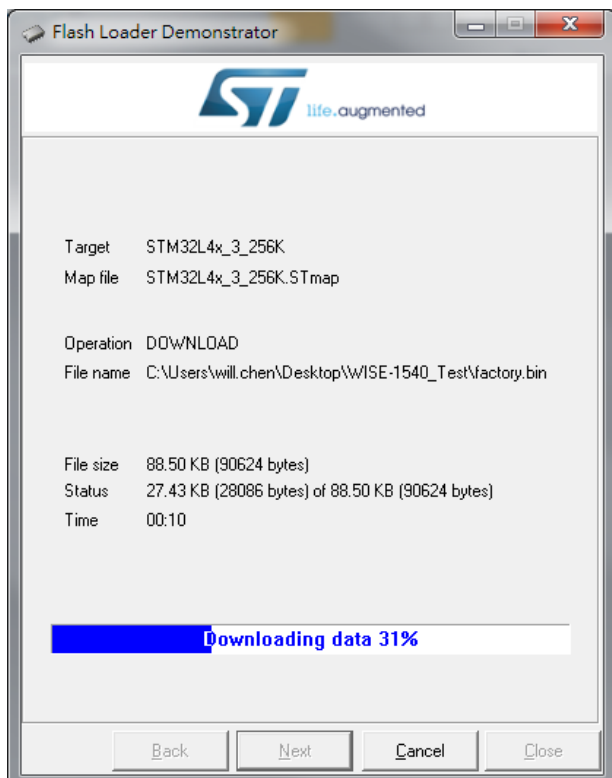
Step04: Select target as below for programming.



Step05: Enable option "Global Erase" and programming for binary from the path you specified.



Step06: Programming in progress.



Step07: Check the process of programming is done and successful.



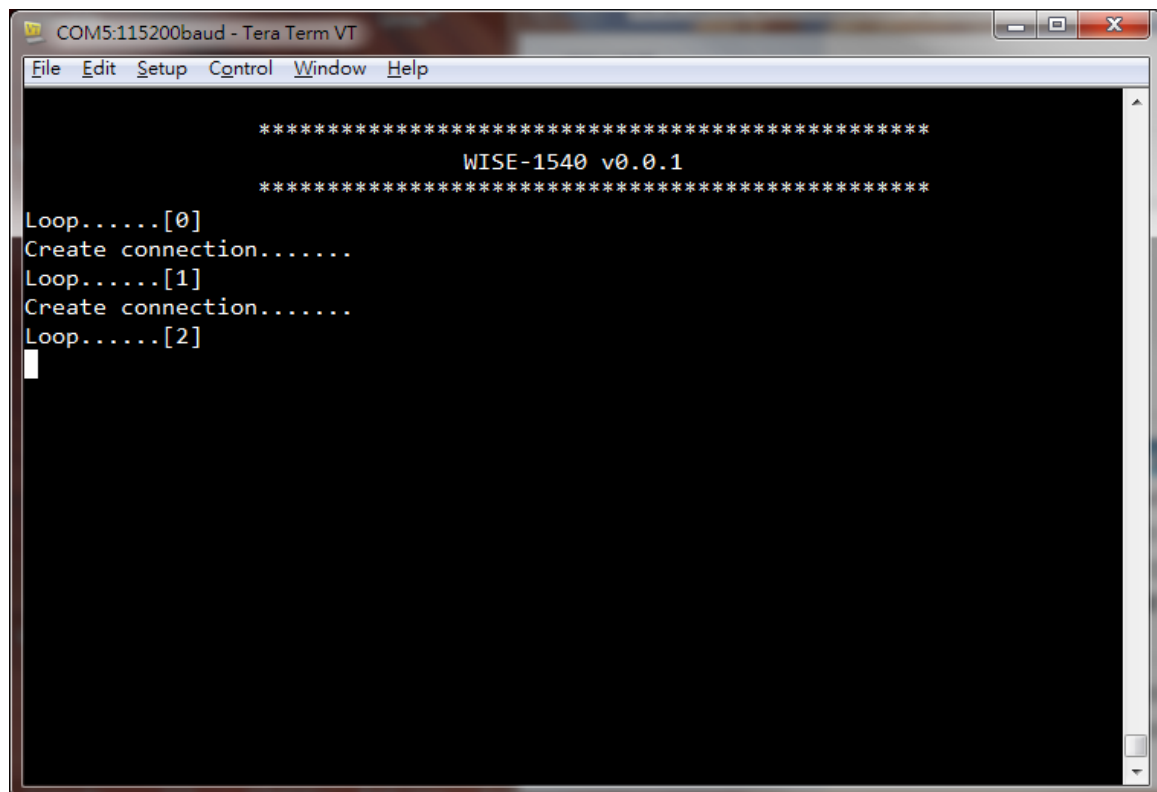
Step08 Set into boot mode.

Please check pin1 of SW2 to "OFF" on WISE-ED20.



Step09: Boot from flash.

Press reset button and you can see snapshot like as below.



The screenshot shows a Tera Term VT window with the title bar 'COM5:115200baud - Tera Term VT'. The menu bar includes 'File', 'Edit', 'Setup', 'Control', 'Window', and 'Help'. The main window area displays the following text:

```
*****  
WISE-1540 v0.0.1  
*****  
Loop.....[0]  
Create connection.....  
Loop.....[1]  
Create connection.....  
Loop.....[2]  
|
```

4. Examples

There are two examples in WISE-1540 SDK. One is “Send Data” the user can refer to it for how to use connectivity to send data from mote to gateway. The other is “Exchange Data with Sensor Format” the user can refer to it for how to add sensor in mote and send data accessed from sensor to WISE-PaaS through mote and gateway.

4.1. Compile example

Please refer to WISE-1540-SDK for compilation.

<https://github.com/ADVANTECH-Corp/WISE-1540-SDK>

4.2. Send Data

This example is demonstrated how we could send data from mote to gateway. The user needs to run two programs synchronously that one is on WISE-1540 and the other is on gateway. In example it communicates with gateway using API in “mote_api.h” that can found located at folder “inc\wsn\sm_ipmt” of source tree of WISE-1540 SDK. The API has been implemented based on SmartMesh IP Mote API Guide and the user can refer to it for getting detailed if needed.

Running example on Gateway

Step01: Get IoT Gateway SDK and Setup on PC.

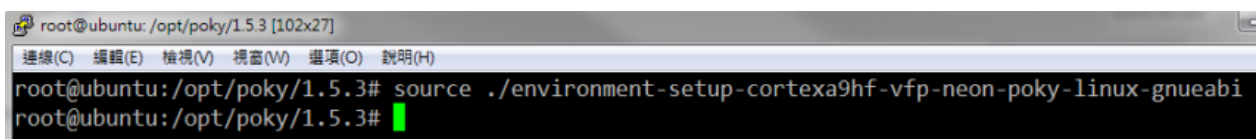
The IoT Gateway SDK is not provided as default and the user needs contact with Advantech to get it.

Step02: Compile example.

① Set environment variable for compilation.

PC # cd /opt/poky/1.5.3/

PC # source ./environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi



```
root@ubuntu: /opt/poky/1.5.3 [102x27]
root@ubuntu:/opt/poky/1.5.3# source ./environment-setup-cortexa9hf-vfp-neon-poky-linux-gnueabi
root@ubuntu:/opt/poky/1.5.3#
```

② The example “wise1021_senddata_gw.tar.bz2” is located in folder “src\sample” of

source tree of WISE-1540 SDK. Please extract wise1021_senddata_gw.tar.bz2 to PC.

PC # tar jxvf wise1021_senddata_gw.tar.bz.

```
root@ubuntu: /tmp [102x27]
root@ubuntu:/tmp# tar jxvf wise1021_senddata_gw.tar.bz2
wise1021_senddata_gw/
wise1021_senddata_gw/DN_LICENSE.txt
wise1021_senddata_gw/ipmg/
wise1021_senddata_gw/ipmg/rs232.c
wise1021_senddata_gw/ipmg/wise1021_receive_sample.c
wise1021_senddata_gw/ipmg/IpMgWrapper.c
```

③ Compile and output binary.

PC # cd wise1021_senddata_gw/

PC # make clean

PC # make

```
root@ubuntu: /home/will/wsn_send_test_for_1021/wise1021_senddata_gw [102x27]
root@ubuntu:/home/will/wsn_send_test_for_1021/wise1021_senddata_gw# make
set -e; for d in sm_clib ipmg; do make -C ${d}; done
make[1]: Entering directory `/home/will/wsn_send_test_for_1021/wise1021_senddata_gw/sm_clib'
arm-poky-linux-gnueabi-gcc -march=armv7-a -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a9 --sysroot=/opt/poky/1.5.3/sysroots/cortexa9hf-vfp-neon-poky-linux-gnueabi -O2 -pipe -g -feliminate-unused-debug-types -c dn_hdlc.c -o dn_hdlc.o
arm-poky-linux-gnueabi-gcc -march=armv7-a -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a9 --sysroot=/opt/poky/1.5.3/sysroots/cortexa9hf-vfp-neon-poky-linux-gnueabi -O2 -pipe -g -feliminate-unused-debug-types -c dn_ipmg.c -o dn_ipmg.o
make[1]: Leaving directory `/home/will/wsn_send_test_for_1021/wise1021_senddata_gw/sm_clib'
```

PC # ls ipmg/

```
root@ubuntu:/tmp/wise1021_senddata_gw# ls ipmg/
IpMgWrapper.c  IpMgWrapper.o  rs232.c  rs232.o  wise1021_receive_sample.c
IpMgWrapper.h  Makefile       rs232.h  wise1021_receive_sample  wise1021_receive_sample.o
```

Step03: Upload binary to WISE-3310.

The following commands are for the user reference. These commands are demonstrated how the PC's file system can be mounted through NFS on WISE-3310 and then the user can copy binary to WISE-3310 using command "cp".

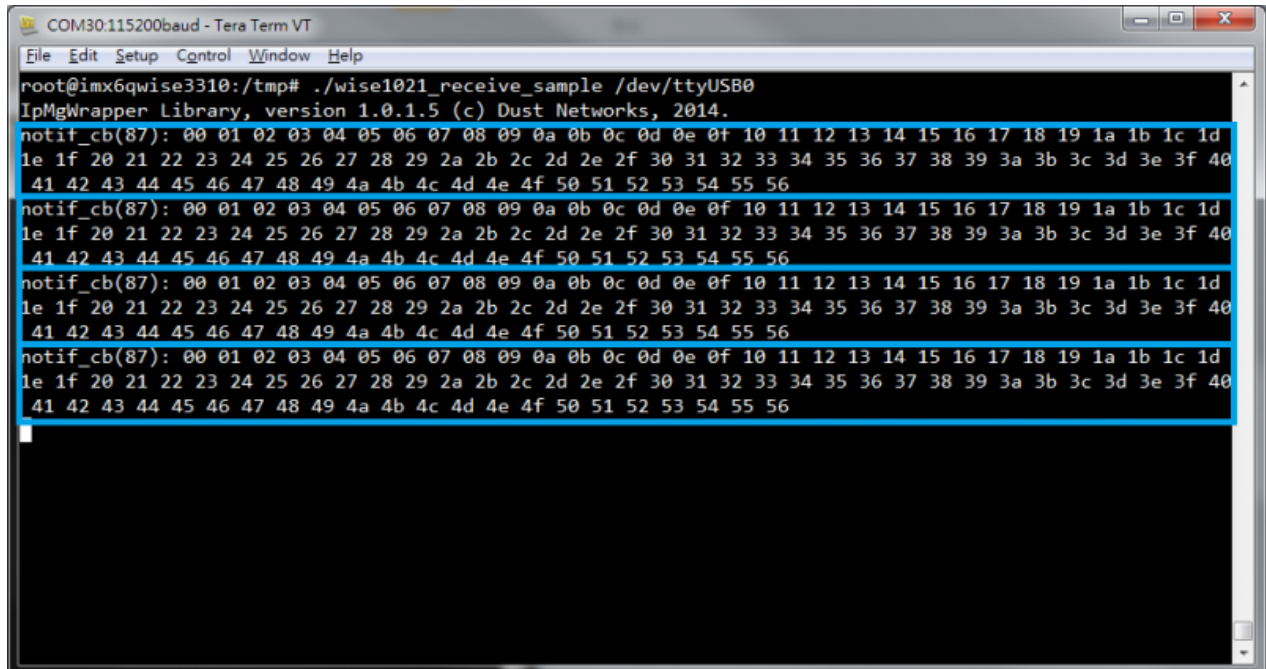
WISE-3310 # mount -t nfs -o nolock 172.22.12.223:/tmp /mnt/

WISE-3310 # cp /mnt/wise1021_senddata_gw/ipmg/wise1021_receive_sample /tmp

Step04: Begin execution.

The user can see snapshot as below while getting data from mote

WISE-3310 # /tmp/ipmg/wise1021_receive_sample /dev/ttyUSB0



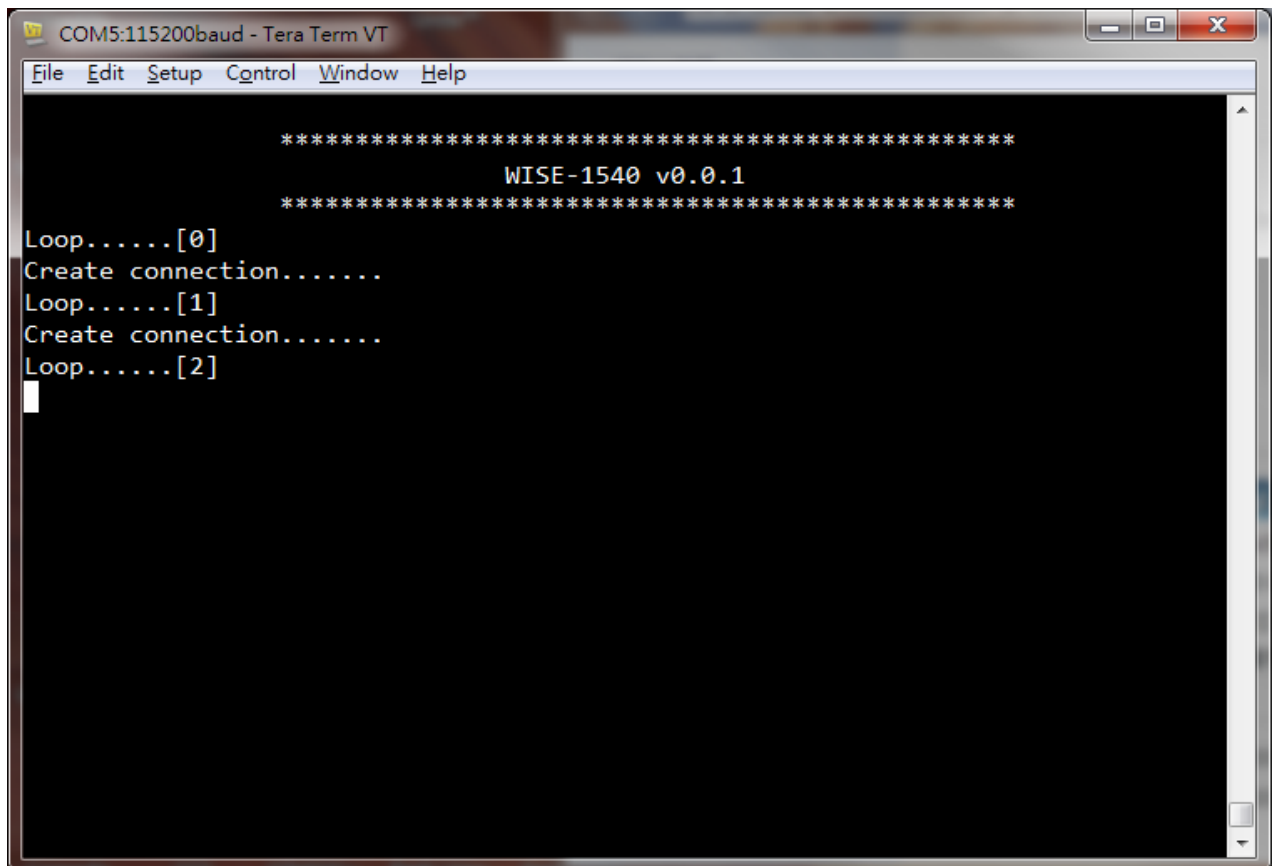
```
COM30:115200baud - Tera Term VT
File Edit Setup Control Window Help
root@imx6qwise3310:/tmp# ./wise1021_receive_sample /dev/ttyUSB0
IpMgWrapper Library, version 1.0.1.5 (c) Dust Networks, 2014.
notif_cb(87): 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d
1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40
41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56
notif_cb(87): 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d
1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40
41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56
notif_cb(87): 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d
1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40
41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56
notif_cb(87): 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d
1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40
41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56
```

Running example on WISE-1540

Step01: Compile example and programming.

- ① Please refer to [WISE-1540-SDK](#) for compilation.
- ② Please refer to section "[Programming through WISE-ED22](#)" or "[Programming through WISE-ED20](#)" for flash programming.

Step02: Check status.



The screenshot shows a Tera Term VT terminal window with the title bar "COM5:115200baud - Tera Term VT". The menu bar includes "File", "Edit", "Setup", "Control", "Window", and "Help". The terminal output displays the following text:

```
*****  
                WISE-1540 v0.0.1  
*****  
Loop.....[0]  
Create connection.....  
Loop.....[1]  
Create connection.....  
Loop.....[2]  
█
```

4.3.Exchange Data with Sensor Format

This example is demonstrated how it could exchange data with sensor format between mote and gateway. The sensor format defined by Advantech combines IPSO data format with CoAP like protocol. By implementation the user needs to study “WISE sensor API” to know how to add sensor by self. In example it adds fake temperature sensor with sensor format and exchanges data between mote and gateway. Finally the fake temperature will be transfer to WISE-PaaS and it is presented by UI on browser, but WISE-PaaS has no more detailed in this document. Please get contact about WISE-PaaS detailed with Advantech if needed.

How to Handle Data between Mote and Gateway

Based on sensor format, the brief in bidirectional explain how it has been implemented in example as below:

From mote to gateway:

- ① The data accessed from sensor are encoded as packet with sensor format by mote.
- ② The packet is sent from mote to gateway.
- ③ The packet is received and decoded by gateway and then the received information will be transfer to WISE-PaaS.

From gateway to mote :

- ① The data and command are encoded as packet by gateway.
- ② The packet is sent from gateway to mote.
- ③ The packet is received and decoded by mote.
- ④ Depend on command/data received from gateway, the information about WISE-1540 and sensor data in real time are encoded by mote and send it back to gateway.

Running Example

Please refer to the following to run example between mote and gateway:

Running example on Gateway

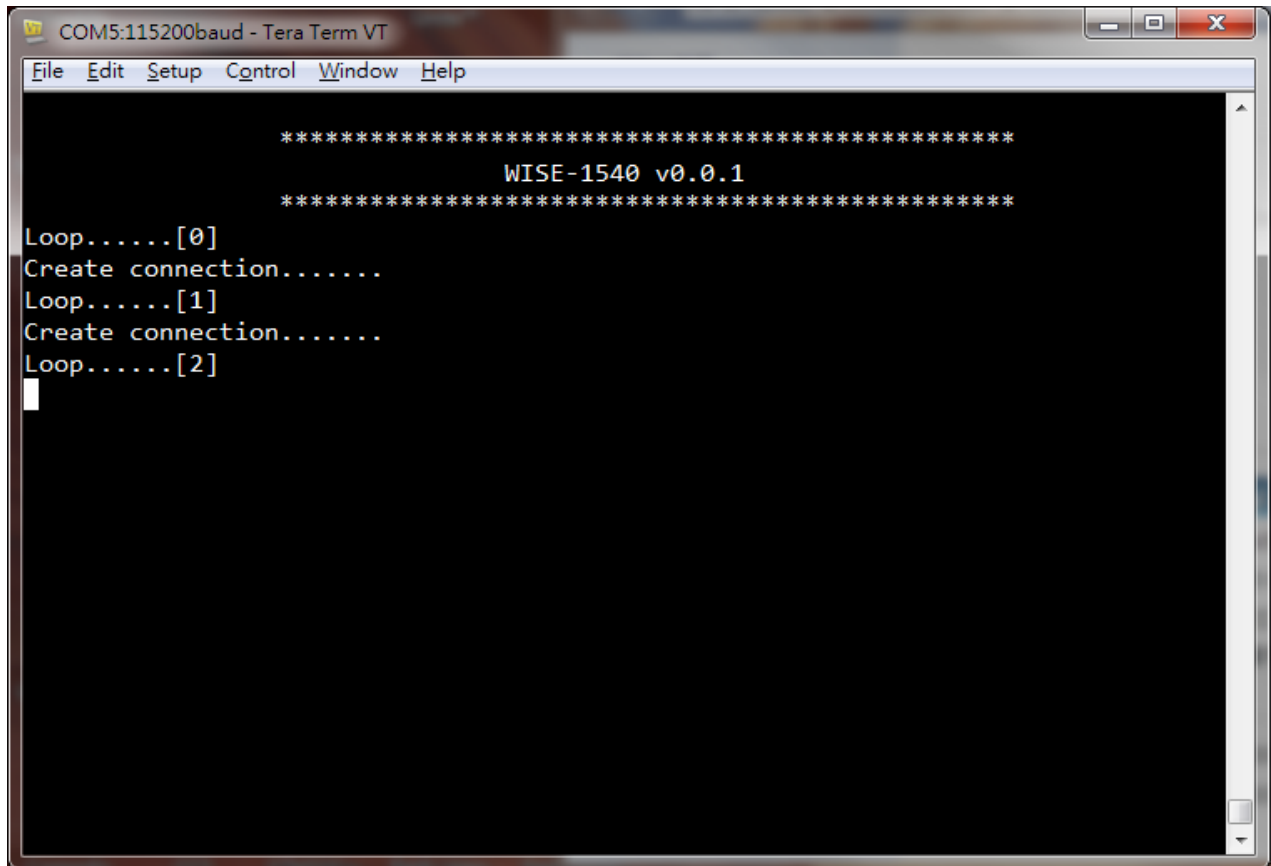
There is nothing to do. This example is supported as default on WISE-3310.

Running example on WISE-1540

Step01: Compile example and programming.

- ① Please refer to [WISE-1540-SDK](#) for compilation.
- ② Please refer to section "[Programming through WISE-ED22](#)" or "[Programming through WISE-ED20](#)" for flash programming.

Step02: Check status.



The screenshot shows a Tera Term VT terminal window titled "COM5:115200baud - Tera Term VT". The window has a menu bar with "File", "Edit", "Setup", "Control", "Window", and "Help". The terminal output displays the following text:

```
*****  
WISE-1540 v0.0.1  
*****  
Loop.....[0]  
Create connection.....  
Loop.....[1]  
Create connection.....  
Loop.....[2]  
|
```

References

Liner Dust

About SmartMesh IP Networks:

- © SmartMesh IP Mote CLI Guide: <http://www.linear.com/docs/41885>
- © SmartMesh IP Mote API Guide: <http://www.linear.com/docs/41886>

Advantech wiki link for WISE-1540:

<http://ess-wiki.advantech.com.tw/view/MCU/WISE-1540>