# User Manual

# WISE-2834

## Intelligent RFID Gateway

ADVANTECH

*Enabling an Intelligent Planet*

# Copyright



# Acknowledgements



## Support

For more information on this and other Advantech products, please visit our websites at: http://www.advantech.com

For technical support and service, please visit our support website at:

http://support.advantech.com/

Part No. 2003283400 Edition 1

Printed in Taiwan November 2019

# 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 merchandize 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

## CE

This product has passed the CE test for environmental specifications when shielded cables are used for external wiring. We recommend the use of shielded cables. This kind of cable is available from Advantech. Please contact your local supplier for ordering information.

## FCC Class A

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

# Technical Support and Assistance

1. Visit the Advantech web site at www.advantech.com/support where you can find the latest information about the product.
2. Contact your distributor, sales representative, or Advantech's customer service center for technical support if you need additional assistance. Please have the following information ready before you call:
   - Product name and serial number
   - Description of your peripheral attachments
   - Description of your software (operating system, version, application software, etc.)
   - A complete description of the problem
   - The exact wording of any error messages

# Safety Precaution - Static Electricity

Follow these simple precautions to protect yourself from harm and the products from damage.

- To avoid electrical shock, always disconnect the power from your PC chassis before you work on it. Don't touch any components on the CPU card or other cards while the PC is on.
- Disconnect power before making any configuration changes. The sudden rush of power as you connect a jumper or install a card may damage sensitive electronic components.

# NCC 警语

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。

# 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.

15. DO NOT LEAVE THIS EQUIPMENT IN AN ENVIRONMENT WHERE THE STORAGE TEMPERATURE MAY GO BELOW -10° C (14° F) OR ABOVE 60° C (140° F). THIS COULD DAMAGE THE EQUIPMENT. THE EQUIPMENT SHOULD BE IN A CONTROLLED ENVIRONMENT.
16. CAUTION: DANGER OF EXPLOSION IF BATTERY IS INCORRECTLY REPLACED. REPLACE ONLY WITH THE SAME OR EQUIVALENT TYPE RECOMMENDED BY THE MANUFACTURER, DISCARD USED BATTERIES ACCORDING TO THE MANUFACTURER'S INSTRUCTIONS.
17. ATTENTION: Danger d'explosion si la batterie est mal REMPLACE. REMPLACER UNIQUEMENT PAR LE MEME TYPE OU EQUIVALENT RECOMMANDÉ PAR LE FABRICANT, jeter les piles usagées SELON LES INSTRUCTIONS DU FABRICANT.
18. The sound pressure level at the operator's position according to IEC 704-1:1982 is no more than 70 dB (A).

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

# Contents


## Chapter 1 Product Overview .....1

1.1 Introduction ..... 2
1.2 Series Family and Specifications ..... 2
1.3 Feature Highlight ..... 2
1.4 Mechanical Design and Dimensions ..... 2
Figure 1.1 WISE-2834 Dimension Front and Side ..... 2
1.5 LED Definition ..... 3
Figure 1.2 WISE-2834 LED Indicator ..... 3
1.6 Package Information ..... 3

## Chapter 2 Product Specification .....5

2.1 General Specifications ..... 6
2.1.1 RFID Communication ..... 6
2.1.2 System Hardware ..... 6
2.1.3 Communication ..... 7
2.1.4 I/O ..... 7
2.1.5 Software ..... 7
2.1.6 Configuration Interface ..... 8
2.1.7 Pin Assignment ..... 8
Figure 2.1 WISE-2834 Pin Assignment ..... 8
2.1.8 Application Wiring ..... 8
Figure 2.2 WISE-2834 Digital Input Dry Contact Wiring Diagram 8
Figure 2.3 WISE-2834 Digital Input Wet Contact Wiring Diagram 9
Figure 2.4 WISE-2834 Digital Output Wiring Diagram ..... 9
2.1.9 Block Diagram ..... 10
Figure 2.5 WISE-2834 Block Diagram ..... 10

## Chapter 3 Mechanical and Hardware Installation 11

3.1 Interface Introduction ..... 12
Figure 3.1 WISE-2834 Interface Introduction ..... 12
3.2 Mounting ..... 12
3.2.1 Wall Mounting ..... 12
Figure 3.2 Wall Mounting Install ..... 12
3.2.2 DIN-Rail Mounting ..... 13
Figure 3.3 DIN Mounting Install ..... 13
Figure 3.4 DIN Mounting_Front ..... 13
Figure 3.5 DIN Mounting_Back ..... 14
3.2.3 Extrusion mount - Vertical ..... 14
Figure 3.6 Extrusion Mount_Vertical_Back ..... 14
Figure 3.7 Extrusion mount_Vertical_Upper ..... 14
Figure 3.8 Extrusion Mount_Vertical_Back ..... 15
Figure 3.9 Extrusion Mount_Vertical_Front ..... 15
3.2.4 Extrusion mount - Horizontal ..... 16
Figure 3.10 Extrusion Mount_ Horizontal_Back ..... 16
Figure 3.11 Extrusion Mount_ Horizontal_Upper ..... 16
Figure 3.12 Extrusion Mount_ Horizontal_Back ..... 16
Figure 3.13 Extrusion mount_ Horizontal_Front ..... 17
3.3 mPCIe Card ..... 17

Figure 3.14 mPCIe Card Location ..... 17
3.4 Power Supply Wiring ..... 17

Chapter 4 System Configuration ..... 19

4.1 Connection ..... 20
Figure 4.1 WISE-2834 Connection_WISE Studio 1 ..... 20
Figure 4.2 WISE-2834 Connection_WISE Studio 2 ..... 20
Figure 4.3 WISE-2834 Connection_WISE Studio 3 ..... 21
Figure 4.4 WISE-2834 Web Portal ..... 21
4.2 Web utility ..... 22
4.2.1 Configuration module name ..... 22
4.2.2 Network setting ..... 22
4.2.3 Date/time, time zone settings ..... 23
4.2.4 System restart ..... 23
4.2.5 Watch dog enable/disable ..... 23
4.2.6 I/O firmware download ..... 24
4.2.7 Configuration file upload/export ..... 24
4.2.8 Change password ..... 24
4.3 RFID Antenna setting ..... 25
4.3.1 RFID region setting ..... 25
4.3.2 RFID Antenna Configuration ..... 25
4.3.3 RFID tag filter settings ..... 27
4.3.4 RFID advanced setting and troubleshooting ..... 27
4.4 Image update ..... 28
Figure 4.5 Image Update_SD card ..... 28

Chapter 5 Software Programming (Node-RED) 29

5.1 Terminology Definition ..... 30
Figure 5.1 ISO 18000-6C Tag Memory Map ..... 30
5.2 System Architecture ..... 31
5.2.1 System Architecture ..... 31
Figure 5.2 System Architecture ..... 31
5.3 Graphic programming with Node-RED ..... 31
5.3.1 Node-RED page ..... 31
Figure 5.3 Node-RED Page ..... 31
Figure 5.4 Node-RED sample ..... 32
5.3.2 Tag Inventory ..... 32
5.3.3 Tag Read ..... 34
5.3.4 Tag Write ..... 36
5.3.5 Tag Lock ..... 38
5.3.6 Tag Kill ..... 40
5.3.7 Tag Access Results ..... 41
5.3.8 Get DIO value ..... 43
5.3.9 Get counter value ..... 44
5.3.10 Get counter status ..... 45
5.3.11 Get DO pulse count and continue mode ..... 45
5.3.12 Get latch status ..... 46
5.3.13 Set DO value ..... 46
5.3.14 Set counter value ..... 47
5.3.15 Set DO pulse ..... 48
5.3.16 Clear latch ..... 49
5.4 API for Development ..... 50
5.4.1 RFID APIs ..... 50
5.4.2 I/O APIs ..... 52

Appendix A RFID node output .....55

Table A.1: Inventory report ..... 56
Table A.2: Tag access report ..... 59

Appendix B RFID module error code.....63

Table B.1: Error Code Ranges/Module Table ..... 64
Table B.2: Error Code Details ..... 64

Appendix C RFID Frequency Channel Tables .....81

C.1 United States/Canada/Mexico Region Frequency Channel Table ..... 82
Table C.1: Frequency Channel Table of US Band ..... 82
C.2 Europe Region Frequency Channel Table (ETSI EN 302 208) ..... 82
Table C.2: Frequency Channel Table of EU Band ..... 82
C.3 Europe2 Region Frequency Channel Table(ETSI EN 300 220) ..... 82
Table C.3: Frequency Channel Table of EU2 Band ..... 82
C.4 Taiwan Region Frequency Channel Table ..... 83
Table C.4: Frequency Channel Table of TW Band ..... 83
C.5 China Region Frequency Channel Table ..... 83
Table C.5: Frequency Channel Table of CN Band ..... 83
C.6 South Korea Region Frequency Channel Table ..... 83
Table C.6: Frequency Channel Table of KR Band ..... 83
C.7 Australia/New Zealand Region Frequency Channel Table ..... 84
Table C.7: Frequency Channel Table of AU/NZ Band ..... 84
C.8 Brazil Region Frequency Channel Table ..... 84
Table C.8: Frequency Channel Table of BR Band ..... 84
C.9 Israel Region Frequency Channel Table ..... 84
Table C.9: Frequency Channel Table of IL Band ..... 84
C.10 India Region Frequency Channel Table ..... 84
Table C.10:Frequency Channel Table of IN Band ..... 84
C.11 Japan Region Frequency Channel Table ..... 85
Table C.11:Frequency Channel Table of JP Band ..... 85
C.12 Japan2 Region Frequency Channel Table (with LBT) ..... 85
Table C.12:Frequency Channel Table of JP2 Band ..... 85

# Chapter 1

## Product Overview

## 1.1 Introduction

WISE-2834 is a RFID Gateway IoT device, which integrated with IoT data acquisition, processing, and publishing functions via Node-RED. Data can be accessed via Ethernet and published to the cloud from anywhere.

## 1.2 Series Family and Specifications

WISE-2834 series support frequency band as below.

| Region | Term Name | WISE-2834-CA | WISE-2834-EA |
|---|---|---|---|
| United States / Canada / Mexico | US / CA / MX | V | V |
| Europe | EU/ETSI EN 302 208 | - | V |
| Europe 2 | EU2 / ETSI EN 300 220 | - | V |
| Taiwan | TW | V | V |
| China | CN | V | - |
| South Korea | KR | V | V |
| Australia / New Zealand | AU / NZ | V | V |
| Brazil | BR | V | V |
| Israel | IL | V | V |
| India | IN | - | V |
| Japan | JP | V | - |

## 1.3 Feature Highlight

- 4-ports UHF RFID Antenna
- 4-ch Digital Input and 4-ch Digital Output
- Ethernet and Wi-Fi interface for up-link
- Graphic program tool by Node-RED for data read/write, filtering and transfer

## 1.4 Mechanical Design and Dimensions

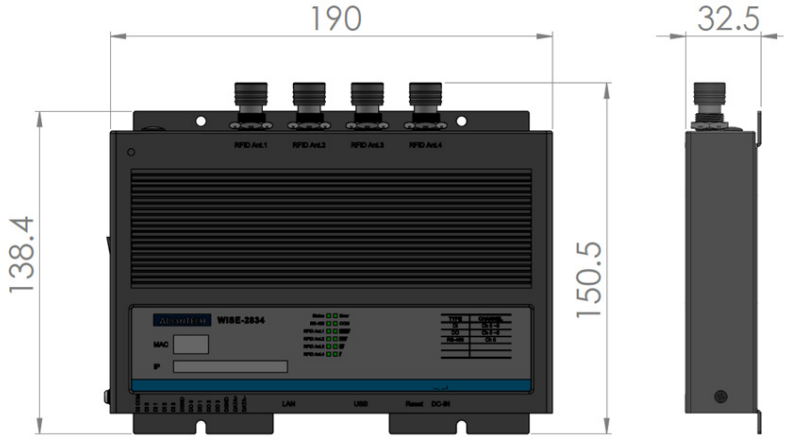



**Figure 1.1 WISE-2834 Dimension Front and Side**

## 1.5 LED Definition

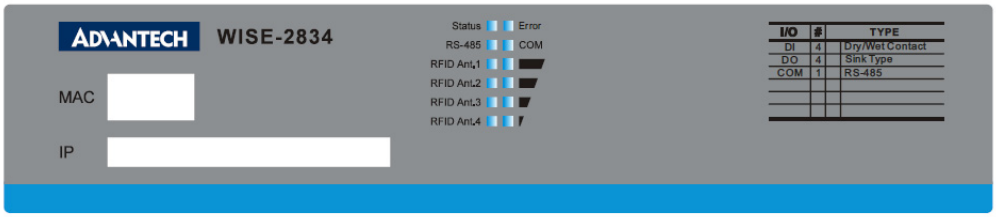
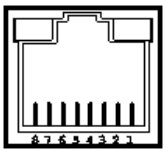



**Figure 1.2 WISE-2834 LED Indicator**

| | LED | Colour | Behaviour | Description |
|---|---|---|---|---|
| **LED Indication** | Status | Green | On/Blink | Light is on when power is on, while the system is starting up the light blinks. |
| | Error | Red | On | System Error |
| | RS-485 | Yellow | Blink | TX data in transmission |
| | | Green | Blink | RX data in receive |
| | COM | Green | On | When enable mPCIe module |
| | | | off | When disable mPCIe module |
| | RFID1#4 | Green | On | RFID channel enable |
| | Signal Strength (when using mPCIe module) | Yellow | On*1~4 | Poor to full signal respectively |
| | | | Off | No Signal/Limited AP Mode |



| RJ45 | Color | Behavior | Description |
|---|---|---|---|
| Left Light | Green | On | 1Gbps connection |
| | Orange | On | 10/100Mbps connection |
| Right Light | Green | Blink | Communication active |

## 1.6 Package Information

- 1 x WISE-2834 module
- 1 x Quick startup manual

# Chapter 2

## Product Specification

# 2.1 General Specifications

## 2.1.1 RFID Communication

| | |
|---|---|
| **RFID Standard** | EPC Global Class 1 Gen. 2 (ISO18000-6C) |
| **Frequency Band** | US: 902.75MHz~927.25MHz<br>EU: 865.7MHz~867.5MHz<br>CN: 920.625MHz~924.375MHz<br>JP: 916.8MHz~920.4MHz |
| **RFID Power Output** | Available to adjust from +10 ~ +31.5dBm |
| **Max Receive Sensitivity** | -74dBm |
| **Antenna Number** | 4 port antennas |
| **Antenna Connector** | 4 RP-TNC |

## 2.1.2 System Hardware

| | |
|---|---|
| **Certification** | CE, FCC, NCC |
| **Power** | 10~50VDC<br>Power consumption:3W (TYP.), 15W (Max.) |
| **Dimension** | 190x120x30.2 mm |
| **CPU** | ARM Cortex-A8, 300MHz<br>ARM Cortex-M0 32-Bit 32MHz |
| **Storage** | NAND Flash 512MB for system |
| **Memory** | DDR3L 512MB |
| **LED Indicator** | Status, Error, Serial (Tx, Rx),Wi-Fi communication, RFID Channel on/off, Wi-Fi Signal Strength |
| **SD Slot** | 1 x Micro SD card |
| **USB Port** | 1 x USB2.0 High Speed (Up to 480Mbps) |
| **Mounting** | DIN 35 rail, Wall, and Pole |
| **Watch Dog Timer** | System & Power Monitor |
| **Real Time Clock** | Time Accuracy to Second (RTC accuracy 2sec/day) |
| **Operating Temperature** | -25°C~ 50°C |
| **Operating Humidity** | 20~95% RH |
| **Storage Temperature** | -40°C~ 85°C |
| **Storage Humidity** | 0~95% RH |

### 2.1.3 Communication

| | |
|---|---|
| **Ethernet** | 1 x 10/100 Based-T RJ-45 |
| **Serial Port**<br>**Isolation** | 1 x RS-485: 300 ~ 115.2k bps<br>3KV rms |
| **Wireless (Optional)** | Interface: 1x Mini-PCIe (Half-size)<br>Type: WiFi |

### 2.1.4 I/O

| | | |
|---|---|---|
| **Digital Input** | **Channel** | 4 |
| | **Max. Input current** | 40mA |
| | **Isolated voltage** | 2kV |
| | **Counter input** | 3kHz |
| | **Dry Contact** | |
| | **Logic 0** | Close to GND |
| | **Logic 1** | Open |
| | **Wet Contact** | |
| | **Logic 0** | 0~3VDC |
| | **Logic 1** | 10~30VDC |
| **Digital Output** | **Channel** | 4 |
| | **Isolated voltage** | 2kV |
| | **Connection type** | Sink |
| | **Supply voltage** | 0~30 VDC |
| | **Max. output current** | 0.4A / channel |
| | **Pulse output** | Up to 5KHz |
| | **On-state resistance (Tj=25°C)** | 550mΩ |
| | **Protection** | Over load, over temperature & short circuit. |

### 2.1.5 Software

| | |
|---|---|
| **Configuration Tool** | WISE Studio |
| **Programming** | Node-RED, Linux OS |

***Note!*** *WISE-2834 modules can operate below 30% humidity. However, environments with low relative humidity are prone to problems with electrostatic discharge. Therefore, you should ensure that you take adequate precautions by using ground straps, anti-static floor coverings, or similar equipment whenever you handle this equipment, especially in low-humidity environments.*

### 2.1.6 Configuration Interface

- **Interface:** LAN port
- **Connector:** RJ45

### 2.1.7 Pin Assignment

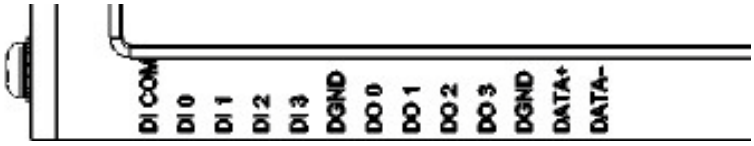



**Figure 2.1 WISE-2834 Pin Assignment**

### 2.1.8 Application Wiring

DI Application Wiring

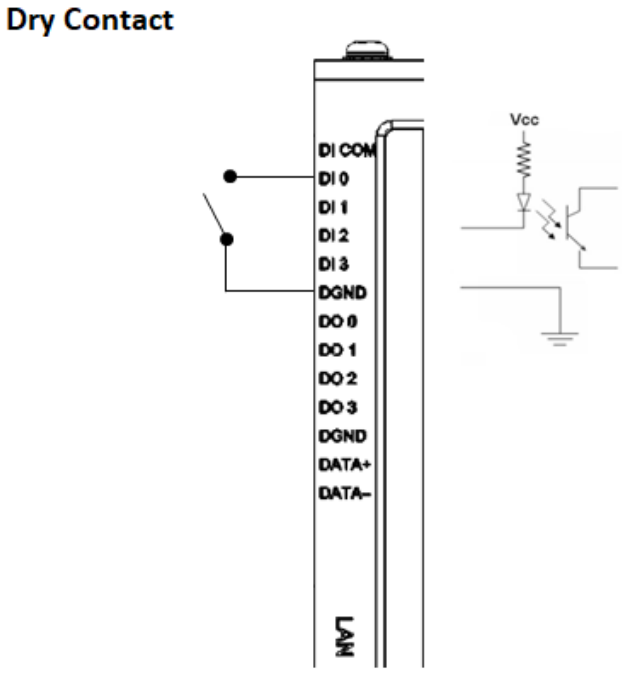



**Figure 2.2 WISE-2834 Digital Input Dry Contact Wiring Diagram**

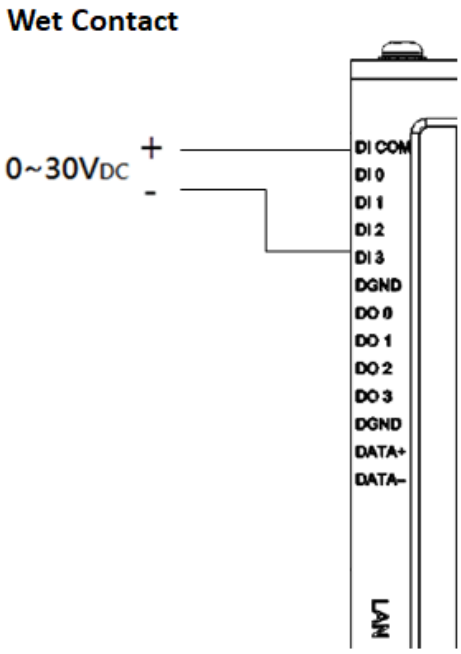



**Figure 2.3 WISE-2834 Digital Input Wet Contact Wiring Diagram**

DO Application Wiring

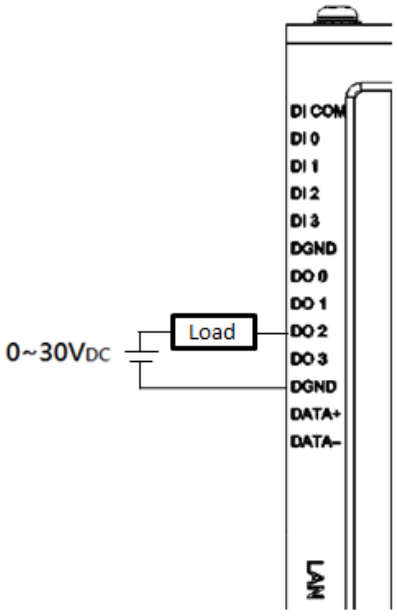



**Figure 2.4 WISE-2834 Digital Output Wiring Diagram**

## 2.1.9 Block Diagram

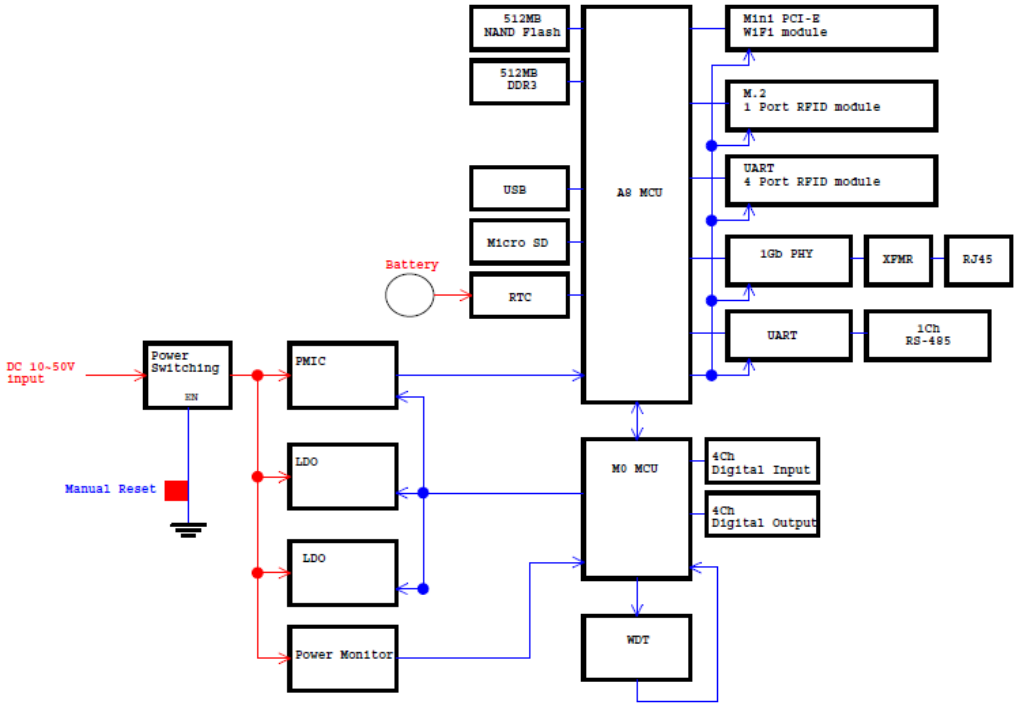



**Figure 2.5 WISE-2834 Block Diagram**

# Chapter 3

## Mechanical and Hardware Installation

## 3.1 Interface Introduction

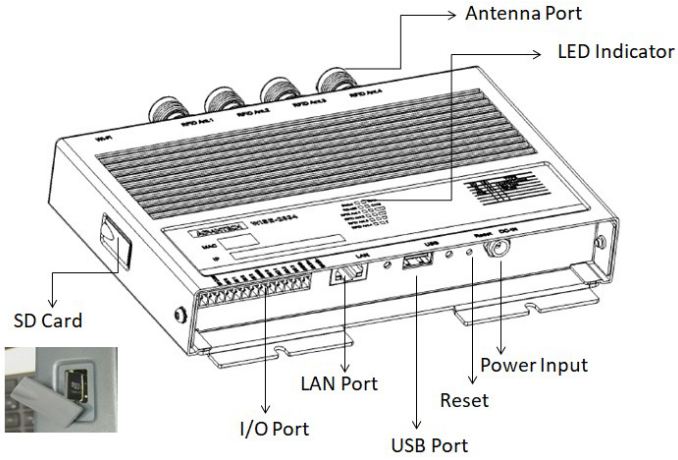



**Figure 3.1 WISE-2834 Interface Introduction**

## 3.2 Mounting

Applicable installation methods are briefly described in the following sections.

### 3.2.1 Wall Mounting

The four screws are installed on wall, panel, or cabinet with WISE-2834.

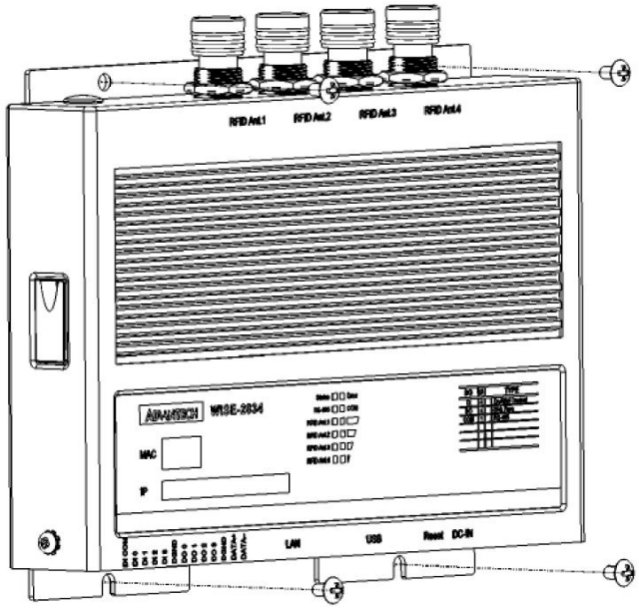


**Figure 3.2 Wall Mounting Install**

### 3.2.2 DIN-Rail Mounting

WISE-2834 can be fixed to a cabinet with mounting rails. Use a screwdriver to fasten the DIN rail adapter to your module. You can then use the end brackets included in the package in order to keep it from sliding.

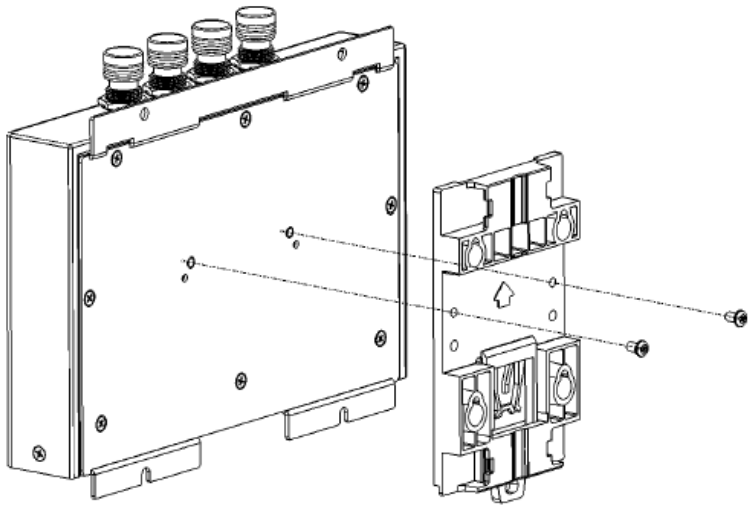


**Figure 3.3 DIN Mounting Install**

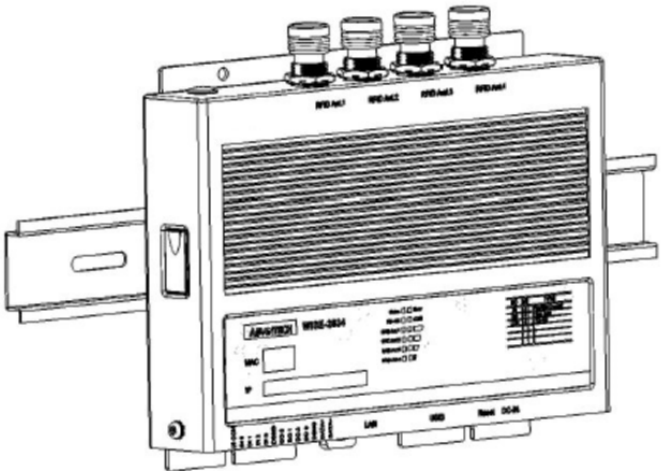


**Figure 3.4 DIN Mounting_Front**

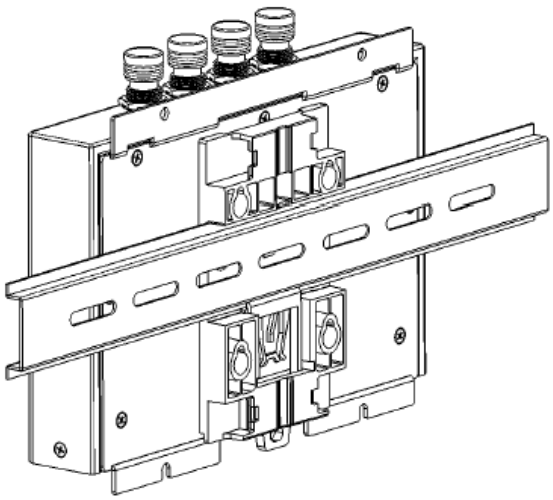


**Figure 3.5 DIN Mounting_Back**

### 3.2.3 Extrusion mount - Vertical

Use a screwdriver to fasten the Extrusion-mount kit to your module.

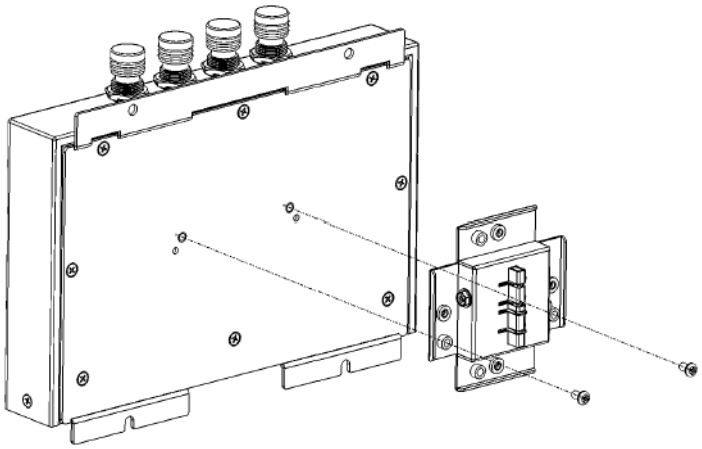


**Figure 3.6 Extrusion Mount_Vertical_Back**

Insert the metal slip of extrusion-mount kit to the seal of extrusion frame, and fasten the screws in left and right side.

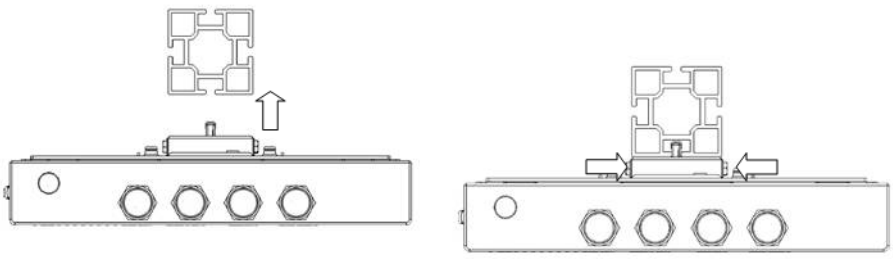


**Figure 3.7 Extrusion mount_Vertical_Upper**

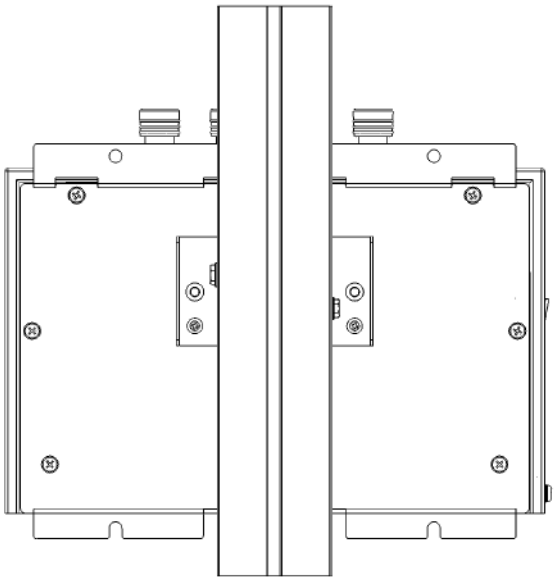


**Figure 3.8 Extrusion Mount_Vertical_Back**

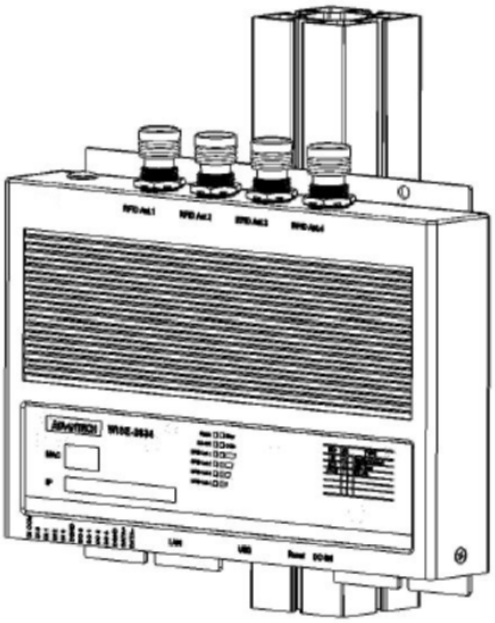


**Figure 3.9 Extrusion Mount_Vertical_Front**

### 3.2.4 Extrusion mount - Horizontal

Use a screwdriver to fasten the Extrusion-mount kit to your module.

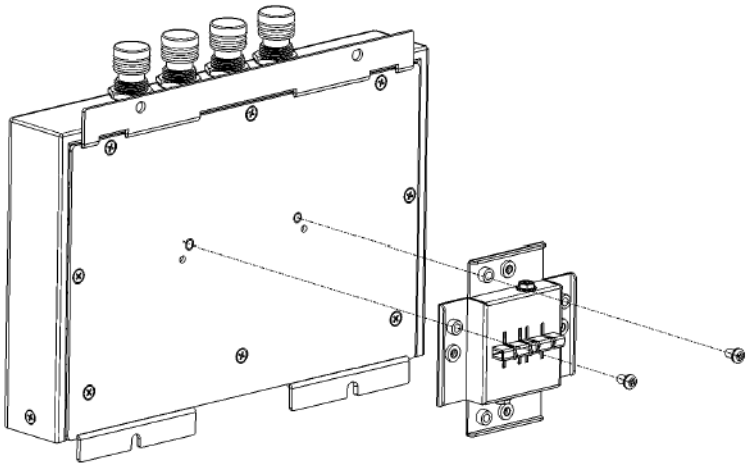


**Figure 3.10 Extrusion Mount_ Horizontal_Back**

Insert the metal slip of the extrusion-mount kit to the seal of extrusion frame, and fasten the screws on the left and right side.

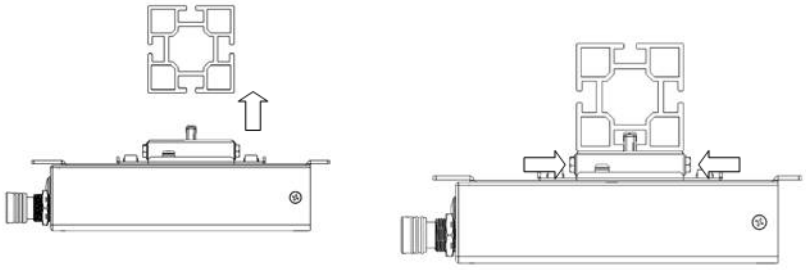


**Figure 3.11 Extrusion Mount_ Horizontal_Upper**

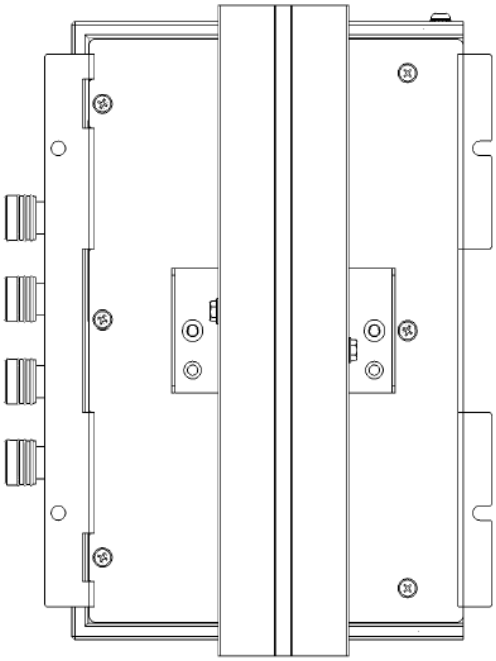


**Figure 3.12 Extrusion Mount_ Horizontal_Back**

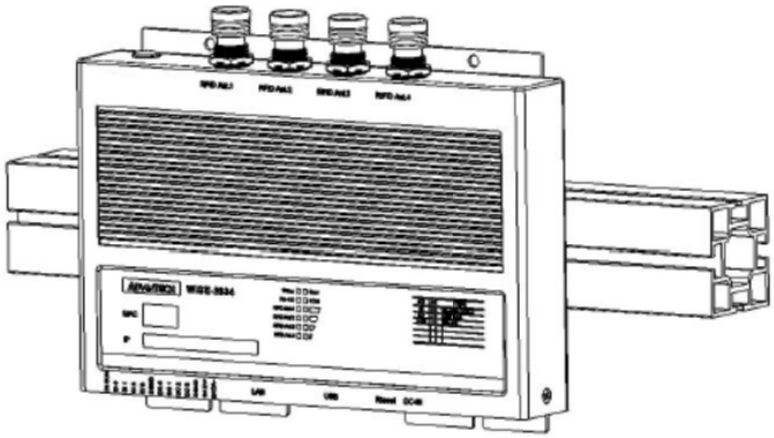


**Figure 3.13 Extrusion mount_ Horizontal_Front**

## 3.3 mPCIe Card

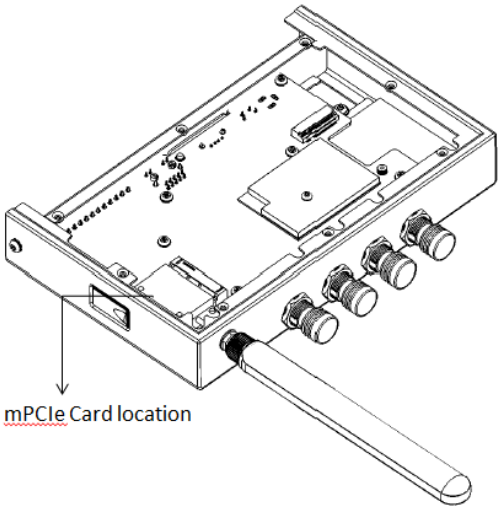



**Figure 3.14 mPCIe Card Location**

## 3.4 Power Supply Wiring

The WISE-2834 is designed for a rated voltage 12 VDC adapter. The power consumption is 3W (TYP.), 15W (Max.)

The sizing of power connector is that inner diameter(2.5mm) and outer diameter(5.7mm).

# Chapter 4

## System Configuration

## 4.1 Connection

1. Plug in a rated voltage 10~50 VDC adapter
2. Connect the module to your computer via the Ethernet port
   The Status light of nameplate LED is on when it's power on. After system start up, the light turns to blink
3. Open WISE Studio and press Go To Configuration


**Figure 4.1 WISE-2834 Connection_WISE Studio 1**

4. Click Connect to link the WISE-2834 and the web configuration page will appear

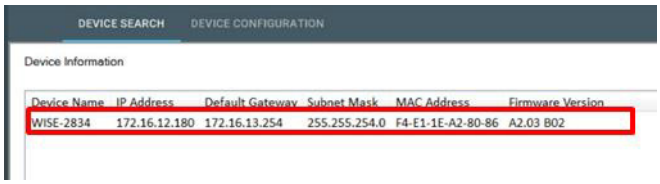



**Figure 4.2 WISE-2834 Connection_WISE Studio 2**

5. Use web configuration in WISE Studio or click Open In Browser to open the web configuration in any browser (Google Chrome is recommended)

- Default account
  - user name: root
  - password: 00000000
- Network: Static/DHCP mode
  If the module cannot receive assigned IP in DHCP mode, the default IP would be 10.0.0.1

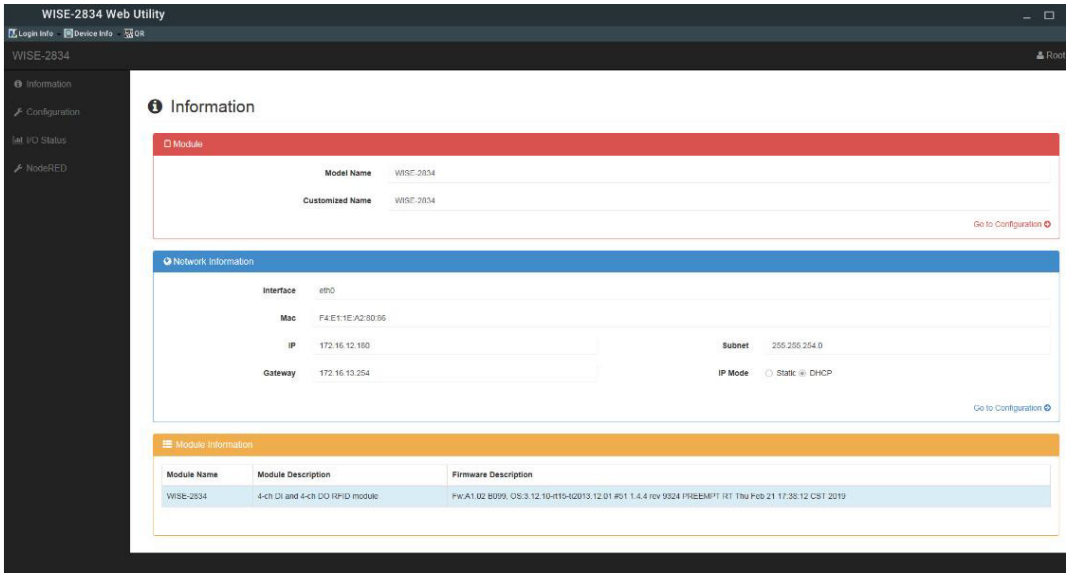



**Figure 4.3 WISE-2834 Connection_WISE Studio 3**

6. If use http://IP , the following figure would be result. Click on "Link", it will re-direct to the correct web page

- Configuration page: https://IP:1880/config
- Node-RED programming page: https://IP:1880


**Figure 4.4 WISE-2834 Web Portal**

- First time log-in after open a browser, it will show "insecure connection"
  - Because this certification is not authenticated by a CA authority
- Click on "advance" and go to the IP link

## 4.2 Web utility

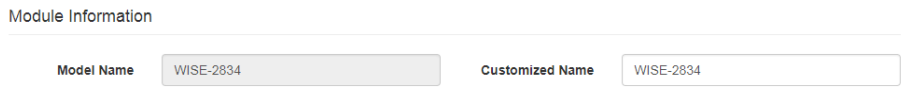
URL: https://IP:1880/config/

Default account

- user name: root
- password: 00000000

### 4.2.1 Configuration module name

Modify Customized Name and click Submit



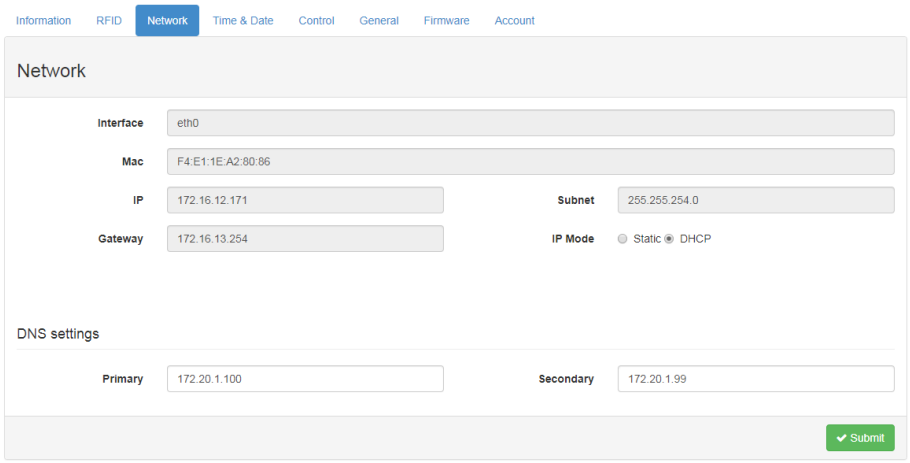

### 4.2.2 Network setting

If you want to change IP, choose Network and then click IP mode

- Static: Please fill in the IP address, subnet mask and gateway IP
- DHCP: No need to modify, as all information will be obtained from the DHCP server

If you choose static mode, we recommend filling in DNS settings

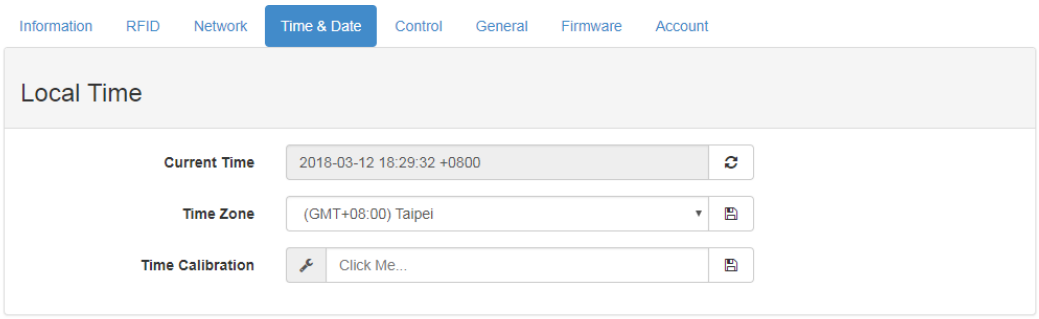
Everything is decided, please press submit button



### 4.2.3 Date/time, time zone settings

Configuration→Time & Date

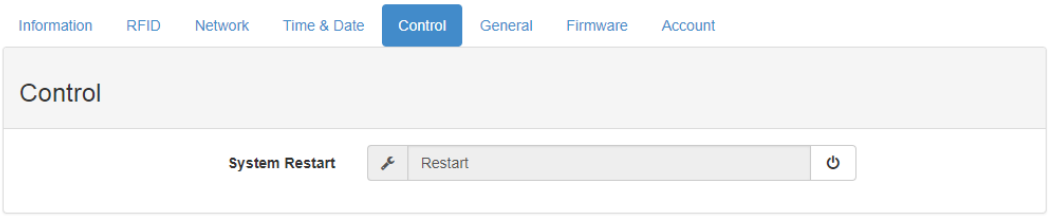
- Current Time
- Time Zone
- Time Calibration




### 4.2.4 System restart

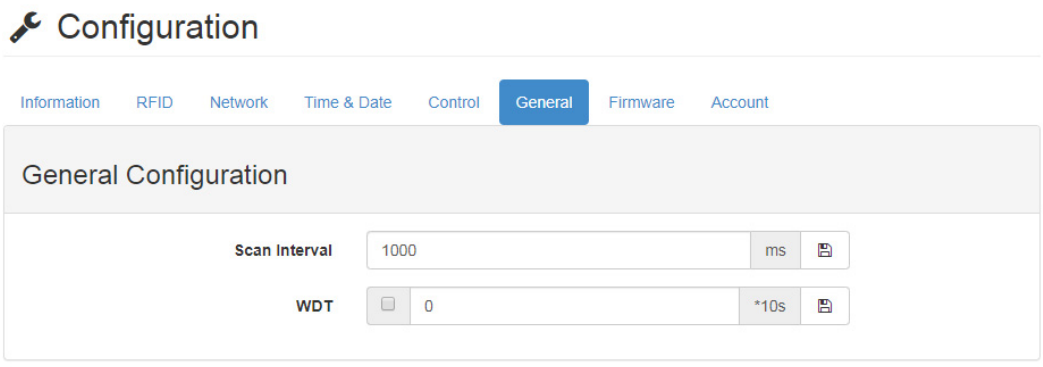
Configuration→Control

Click button to soft-restart system




### 4.2.5 Watch dog enable/disable

Configuration→General




Scan Interval: Frequency update of I/O status

WDT: Enable/disable watch dog function

0: Turn off WDT function

> 0: Turn on WDT function. E.g. inpt 10, 10*10s=100sec → WISE-2834 system don't operate about 100 seconds, and system will restart automatically. WDT function will start in 5 minutes after WISE-2834 power on

### 4.2.6 I/O firmware download

Configuration→Firmware

Choose the I/O firmware binary and click upload firmware button

### 4.2.7 Configuration file upload/export

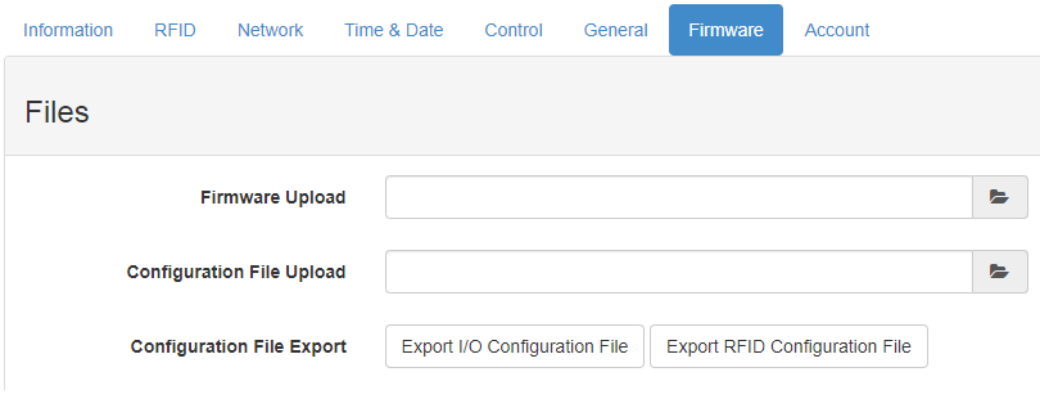
Configuration→Firmware

Upload: Choose the configuration file and click upload file button

Export: Click the Export I/O Configuration File or Export RFID Configuration File button

I/O configuration file name should be "io.cfg"

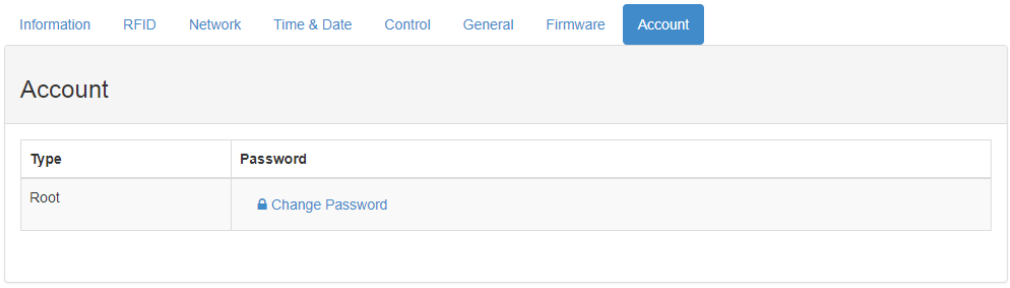
RFID configuration file name should be "rfid.cfg"




### 4.2.8 Change password

Configuration→Account

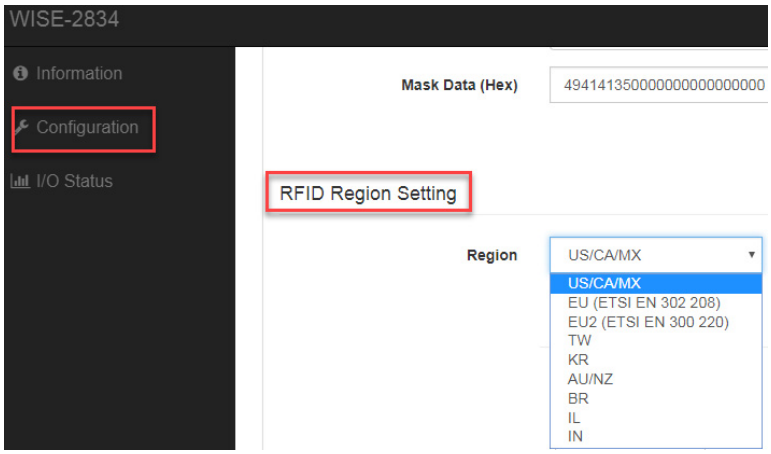
Click change password link to set a new password



# 4.3 RFID Antenna setting

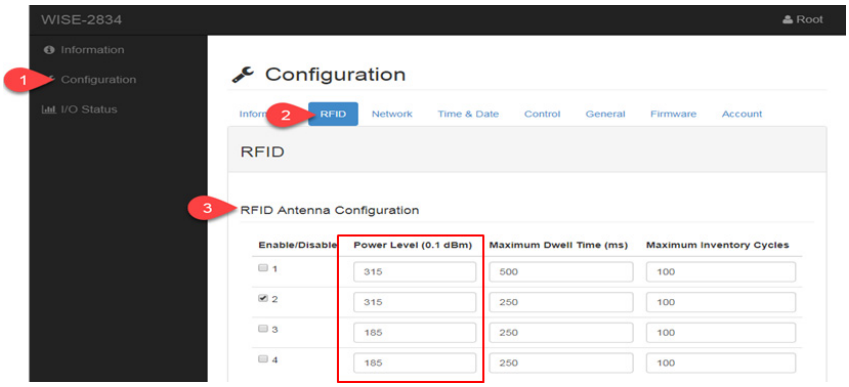
## 4.3.1 RFID region setting

The region can be set according to the country of the end user




## 4.3.2 RFID Antenna Configuration

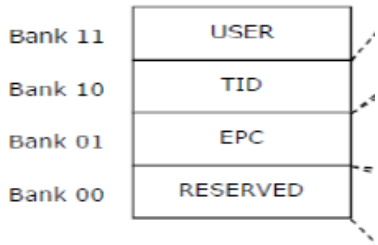
- The channel needs to enable according to the antenna installation
- The power level range is related to the country setting
  - For example: country region is set as US/CA/MX, the range of power lever is between 10~31.5 dBm
- Maximum Dwell Time: Specifies the max amount of time in ms that may be spent on the logical antenna port during a tag-protocol-operation cycle before switching to the next enabled antenna port
- Maximum Inventory Cycles: Specifies the max number of inventory cycles to attempt on the antenna port during a tag-protocol-operation cycle before switching to the next enabled antenna port
- Stop reading condition: The condition when the antenna should stop reading tags
  - Maximum dwell time (ms)
  - Maximum inventory cycles
  - Reader will stop reading tag value if meet 1 of above 2 stop conditions

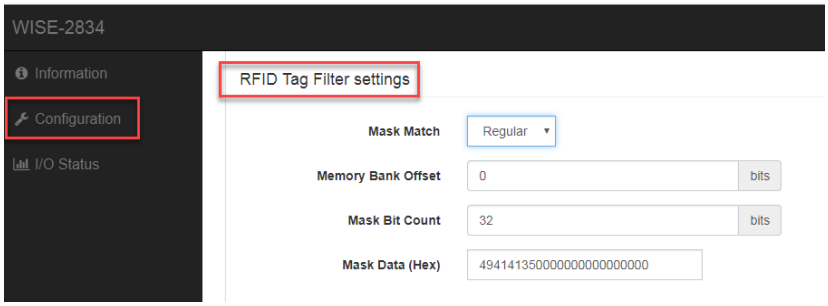


- User can see the LED light in the nameplate, e.g. "RFID1"

### 4.3.3 RFID tag filter settings

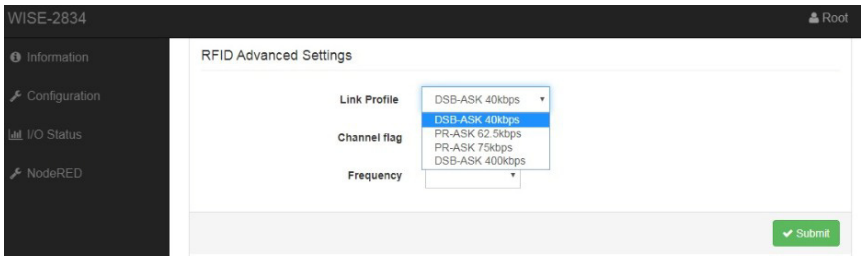
- Filter target: Bank 1, EPC
  - Without CRC and PC.
- Mask match: Determines if the related tag-protocol operation will be applied to tags that match the mask or not
  - 0, Inverse: exclusive the condition
  - 1, Regular: match the condition
- Memory bank offset: the offset in bits, from the start of the EPC of the first bit that will be matched against the mask
- Mask bit count: The number of bits in the mask
- Mask data (Hex): The mapping mask data



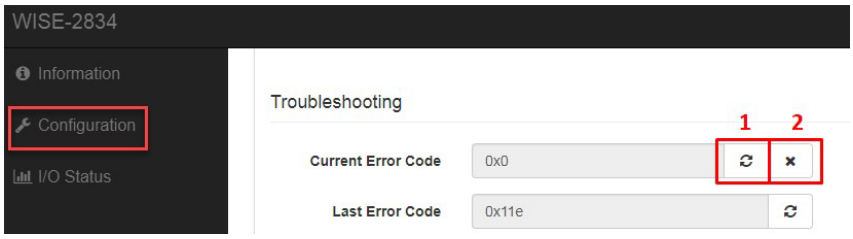




### 4.3.4 RFID advanced setting and troubleshooting

- Link profile: the modulation type and data rate
  - DSB-ASK 40kbps
  - PR-ASK 62.5kbps
  - PR-ADK 75kbps
  - DSB-ASK 400kbps
- Channel flag: Single channel or hopping
  - Hopping: Frequency output uses "hopping" method in the local frequency band
  - Single: Only choose "single" frequency in the local frequency band
- Frequency: the frequency that reader search tags
  The frequency need to be set up if a user select "single" for channel flag



- If error code is not 0x0, then it indicates there are an error occurred during setting or installation




1: Refresh current error code

2: Clear current error code

## 4.4 Image update

- Upload the image files into a micro SD card, image file in https://support.advantech.com/support/new_default.aspx
- Insert the micro SD card into WISE-2834
  - The chip should be face down
  - The words should be face up
- Power-on the module and wait for 10 minutes


**Figure 4.5 Image Update_SD card**

# Chapter 5

## Software Programming (Node-RED)

## 5.1 Terminology Definition

- Tag memory: Tag memory includes Reserved Memory, EPC Memory, Tag Identification (TID) Memory and User Memory.
  - EPC (Electronic Product Code): one common type of data stored in a tag)
  - TID (Tag Identification): TID Memory is the unique tag identifier that cannot be changed or erased. This ID identifies the tag itself, rather than the item it is applied to.
- Reserved Bank: Store Kill Password and Access Password.
- EPC Bank: Store EPC number.
- TID Bank: Tag identifier, each TID number is unique.
- User Bank: Stored data defined by the user.
- Node-RED: A flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs, and online services as part of the Internet of Things.

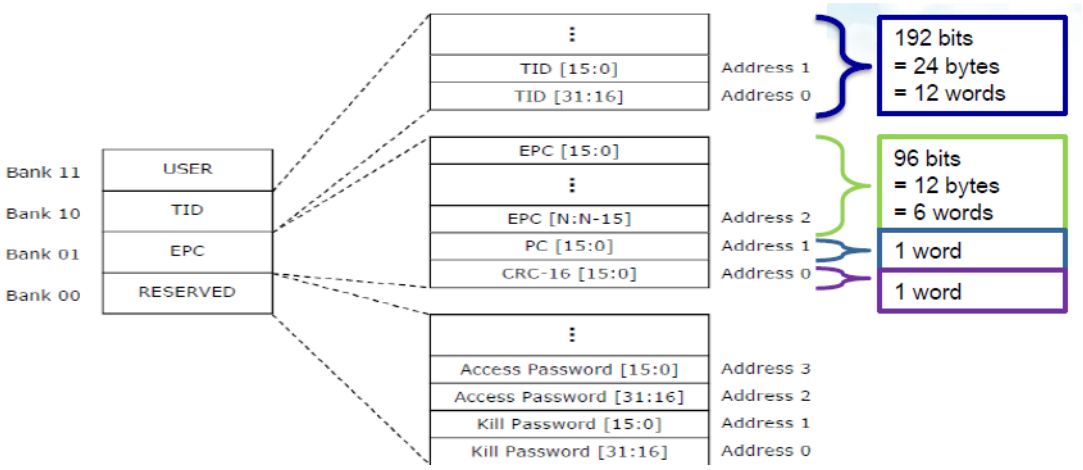



**Figure 5.1 ISO 18000-6C Tag Memory Map**

# 5.2 System Architecture

## 5.2.1 System Architecture

1. System Architecture

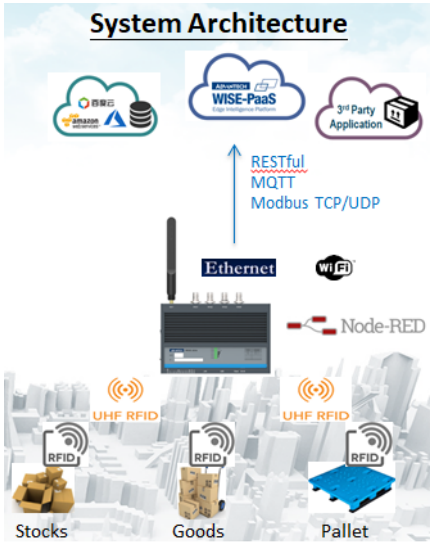



**Figure 5.2 System Architecture**

# 5.3 Graphic programming with Node-RED

## 5.3.1 Node-RED page

- URL: https://IP:1880/
- Default account
  - user name: root
  - password: 00000000

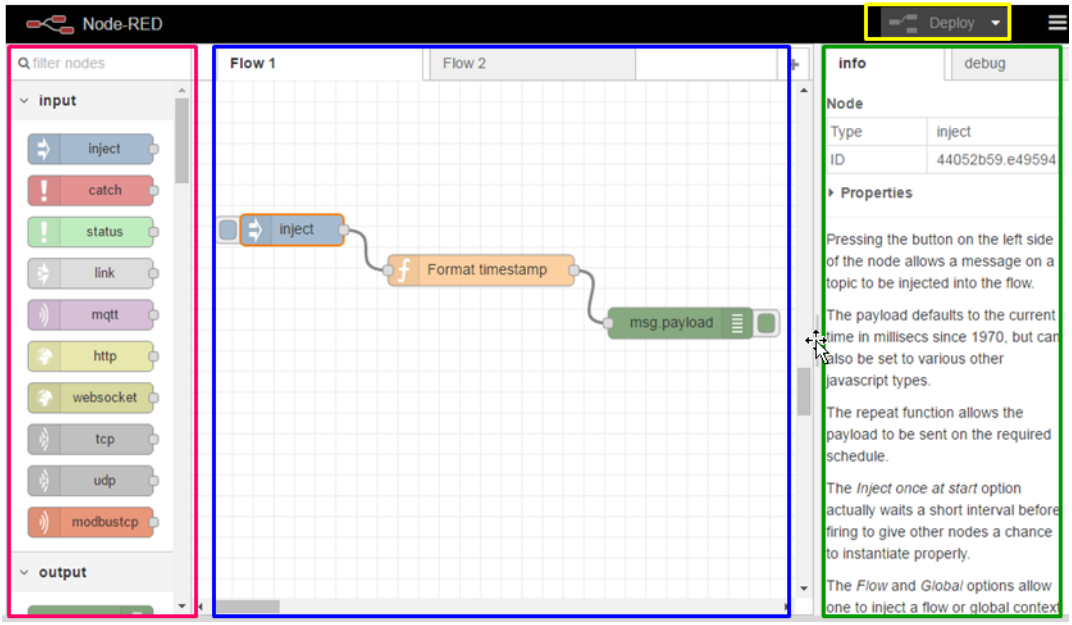



**Figure 5.3 Node-RED Page**

**There are four distinct areas to the Node-RED graphic programming UI**

1. Left panel: Function nodes (called Nodes).
2. Middle area: Graphic programming area where users can drag Nodes to. Each Node has a unique ID and users can graphically program Nodes by linking them.
3. Right panel: Node information and functions.
4. Top toolbar: **Deploy** menu - stores and deploys Node flows on local device.

**Built in examples**

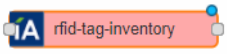
Users can import the built-in example from the internal library.


**Figure 5.4 Node-RED sample**

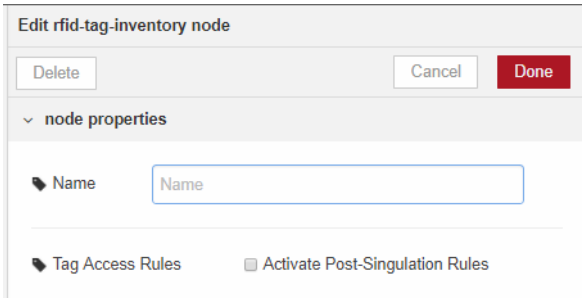
## 5.3.2 Tag Inventory

**Inventory function allows the host to gather EPCs for all tags of interest**

1. Add the **rfid-tag-inventory** node in Node-RED.




2. Fill-in the settings.

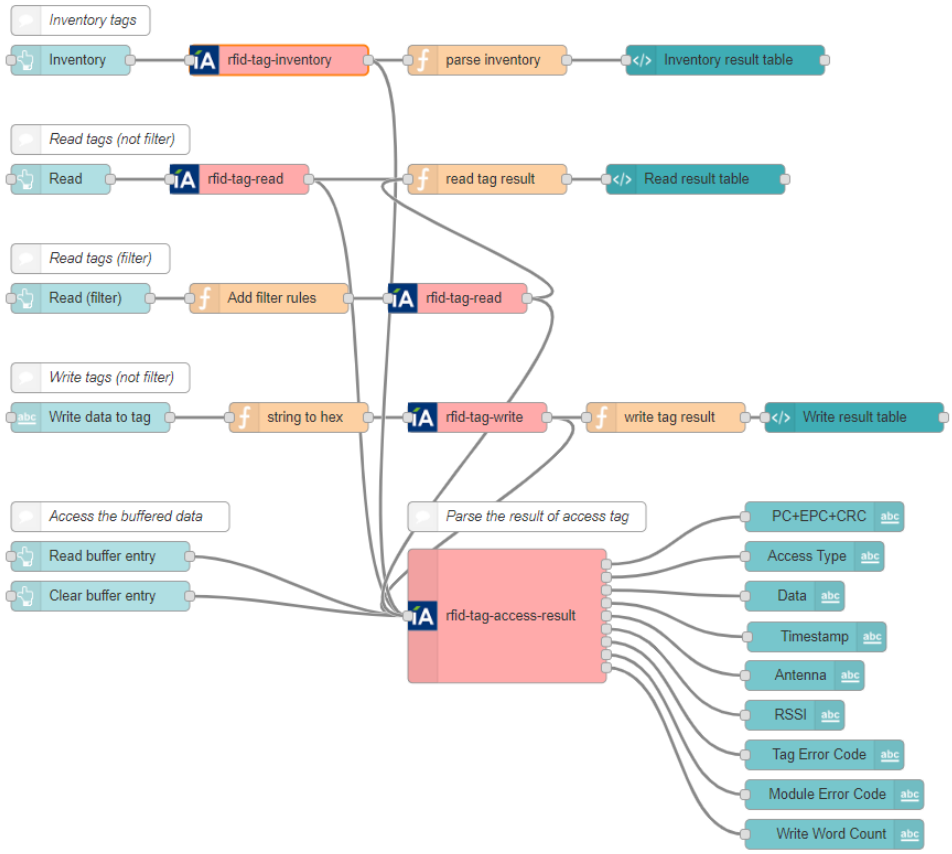



Activate the Post-Singulation Rules: Enable/disable the filter function.

3. Node output.
   The EPC value is included in **msg.Inv.acc_data.**
   For other information please reference the appendix for detailed information.
4. Tag mask setting.
   Enable **Activate Post-Singulation Rules** to filter the tags.

**There are two way to the set filter rule:**

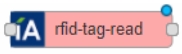
1. Set Tag Filter Setting in the web utility.
2. Send msg.mask to this node.
   Example: msg.mask = { "MaskMatch": "1", "MaskOffset": "0", "MaskCount": "32", "MaskData": "12345678" };
3. Node-RED built in examples.
   Import→Library→Advantech→RFID→Basic_Example.



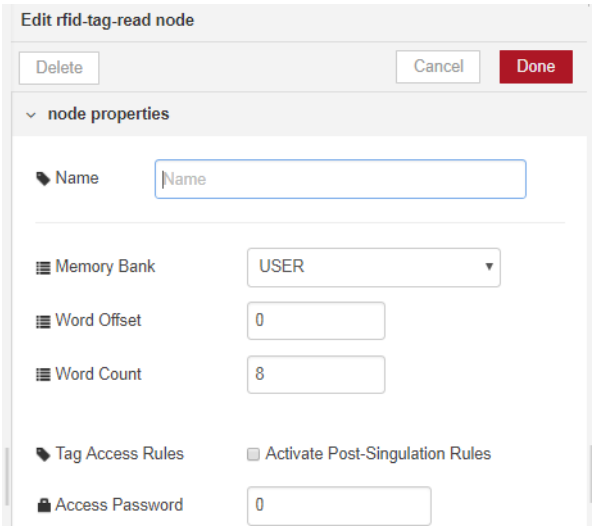
### 5.3.3 Tag Read

**Read tag data according to the memory bank and offset**

1. Add the **rfid tag read** node in Node-RED.




2. Fill-in the settings.

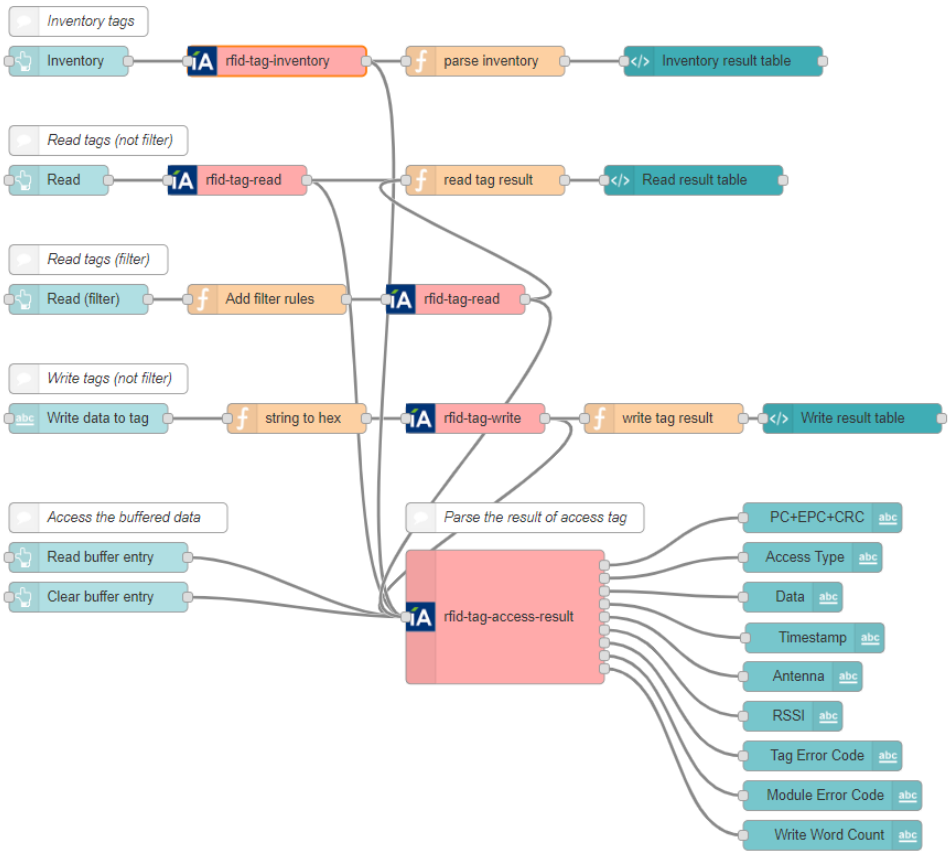



- Memory Bank: EPC/TID/USER/Reserved.
- Word Offset: The offset of the first 16-bit word, zero is the first 16-bit word.
- Word Count: The number of 16-bit words to be read.
- Activate Post-Singulation Rules: Enable/disable the filter function.
- Access Password: Saves the access password for the tags. Zero value indicates no access password.

3. Node output.
   The EPC value is included in **msg.Inv.acc_data.**
   The tag access data is included in **msg.Acc.acc_data.**
   Other information please reference the appendix for detail information.
4. Tag mask setting.
   Enable **Activate Post-Singulation Rules** to filter the tags to be inventory

**There are two way to set filter rule**

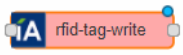
1. Set Tag Filter Setting at web utility.
2. Sends msg.mask to this node.
   Example: msg.mask = { "MaskMatch": "1", "MaskOffset": "0", "MaskCount": "32", "MaskData": "12345678" };
3. Node-RED built in examples.
   Import→Library→Advantech→RFID→Basic_Example.


Inventory tags
Inventory
rfid-tag-inventory
parse inventory
Inventory result table
Read tags (not filter)
Read
rfid-tag-read
read tag result
Read result table
Read tags (filter)
Read (filter)
Add filter rules
rfid-tag-read
Write tags (not filter)
Write data to tag
string to hex
rfid-tag-write
write tag result
Write result table
Access the buffered data
Read buffer entry
Clear buffer entry
Parse the result of access tag
rfid-tag-access-result
PC+EPC+CRC
Access Type
Data
Timestamp
Antenna
RSSI
Tag Error Code
Module Error Code
Write Word Count

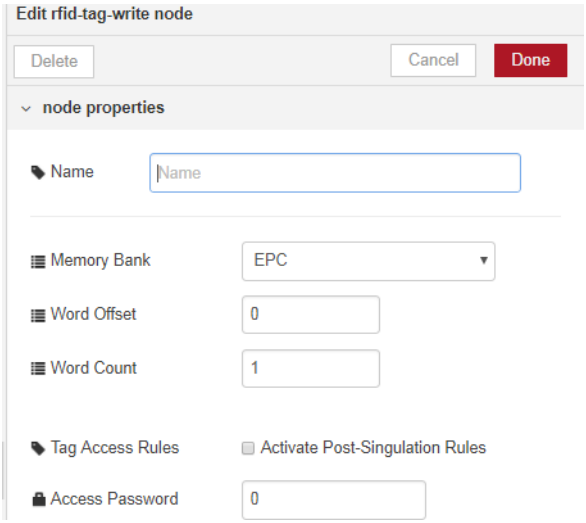
### 5.3.4 Tag Write

**Write tag data according to the memory bank and offset**

1. Add the **rfid tag write** node in Node-RED.



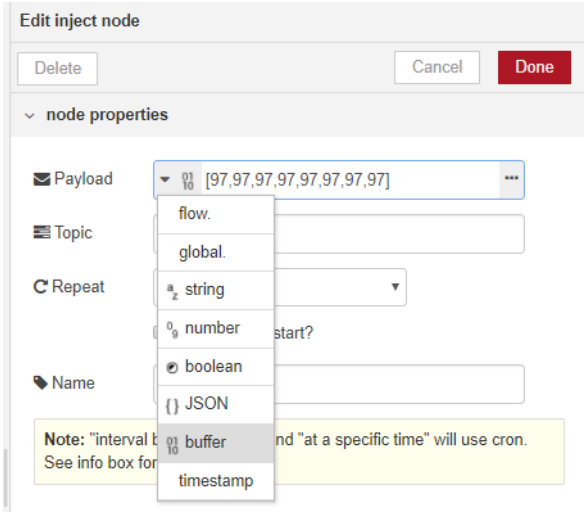

2. Fill in the settings.




- Memory Bank: EPC/TID/USER/Reserved
- Word Offset: The offset of the first 16-bit word, zero is the first 16-bit word.
- Word Count: The number of 16-bit words to be read
- Activate Post-Singulation Rules: Enable/disable the filter function
- Access Password: Saves the access password for the tags. Zero value indicates no access password.

3. Node input.

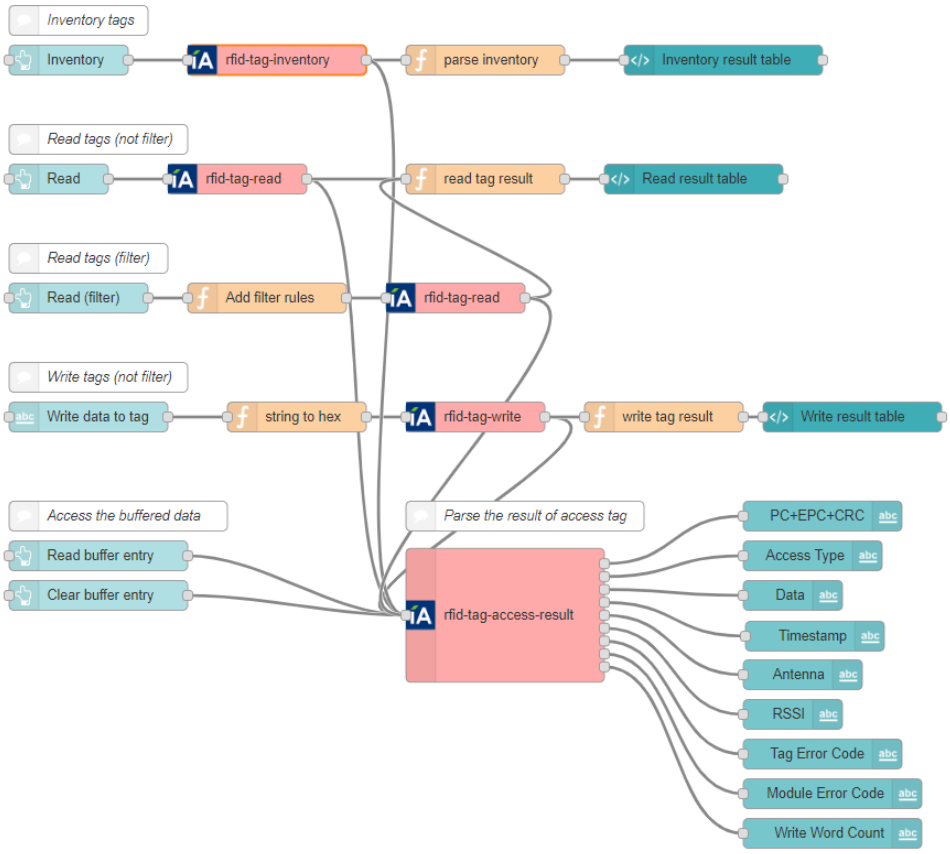
   Node input should be a buffer.
   *For example:*



4. Node output.
   The EPC value is included in **msg.Inv.acc_data.**
   The tag access data is included in **msg.Acc.acc_data.**
   For other information please reference the appendix.
5. Tag mask setting.
   Enable Activate Post-Singulation Rules to filter the tags to be inventory.

**There are two ways to set filter rule**

1. Set Tag Filter Setting at web utility.
2. Send msg.mask to this node.
   Example: msg.mask = { "MaskMatch": "1", "MaskOffset": "0", "MaskCount": "32", "MaskData": "12345678" };
3. Please find Node-RED built-in examples.
   Import→Library→Advantech→RFID→Basic_Example



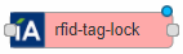
### 5.3.5 Tag Lock

Set the permissions of each bank with a set of tags of interest.

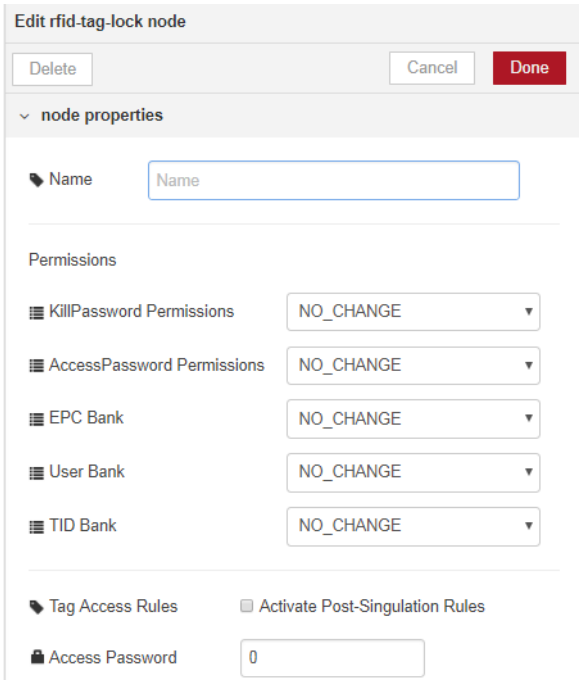
Execute a tag lock for all tags of interest. There are five access permissions that may be set: EPC, TID, user memory banks, and access permissions for the access and kill passwords.

When performing tag-lock operation, the RFID reader uses only the first enabled logical antenna. (i.e. the enabled logical antenna with the smallest logical antenna port number).

1. Add an **rfid tag lock** node in Node-RED.




2. Fill in the settings.




- Kill Password Permissions: The access permissions for the tag kill password.
  - ACCESSIBLE: The password can be read and written when the tag is in either the open or secured states.
  - ALWAYS_ACCESSIBLE: The password can be read and written when the tag is in either the open or secured states, and this access permission should be set permanently.
  - SECURED_ACCESSIBLE: The password can be read and written only when the tag is in the secured states.
  - ALWAYS_NOT_ACCESSIBLE: The password cannot be read or written, and this access permission should be set permanently.
  - NO_CHANGE: The password's access permission should remain unchanged
- Access Password Permissions: The access permissions for the tag access password.
  - ACCESSIBLE: The password can be read and written when the tag is in either the open or secured states.

  - ALWAYS_ACCESSIBLE: The password can be read and written when the tag is in either the open or secured states, and this access permission should be set permanently.
  - SECURED_ACCESSIBLE: The password can be read and written only when the tag is in the secured states.
  - ALWAYS_NOT_ACCESSIBLE: The password cannot be read or written, and this access permission should be set permanently.
  - NO_CHANGE: The password's access permission should remain unchanged.
- EPC Bank: The access permissions for the tag's EPC memory bank.
  - WRITEABLE: The memory bank is writeable when the tag is in either the open or secured states.
  - ALWAYS_WRITEABLE: The memory bank is writeable when the tag is in either the open or secured states, and this access permission should be set permanently.
  - SECURED_WRITEABLE: The memory bank is writeable only when the tag is in the secured states.
  - ALWAYS_NOT_WRITEABLE: The memory bank is not writeable, and this access permission should be set permanently.
  - NO_CHANGE: The memory bank's access permission should remain unchanged.
- User Bank: The access permissions for the tag's User memory bank.
  - WRITEABLE: The memory bank is writeable when the tag is in either the open or secured states.
  - ALWAYS_WRITEABLE: The memory bank is writeable when the tag is in either the open or secured states, and this access permission should be set permanently.
  - SECURED_WRITEABLE: The memory bank is writeable only when the tag is in the secured states.
  - ALWAYS_NOT_WRITEABLE: The memory bank is not writeable, and this access permission should be set permanently.
  - NO_CHANGE: The memory bank's access permission should remain unchanged.
- TID Bank: The access permissions for the tag's TID memory bank.
  - WRITEABLE: The memory bank is writeable when the tag is in either the open or secured states.
  - ALWAYS_WRITEABLE: The memory bank is writeable when the tag is in either the open or secured states, and this access permission should be set permanently.
  - SECURED_WRITEABLE: The memory bank is writeable only when the tag is in the secured states.
  - ALWAYS_NOT_WRITEABLE: The memory bank is not writeable, and this access permission should be set permanently.
  - NO_CHANGE: The memory bank's access permission should remain unchanged.
- Activate Post-Singulation Rules: Enable/disable the filter function.
- Access Password: Saves the access password for the tags. A value of zero indicates no access password. The range is 0x00000000~0xFFFFFFFF.

3. Node output.
   The EPC value is included in **msg.Inv.acc_data**.
   The tag access data is included in **msg.Acc.acc_data**.
   For other information please reference the appendix.
4. Tag mask setting.
   Enable **Activate Post-Singulation Rules** to filter the tags to be inventory.

**There are two way to set filter rule**

1. Set Tag Filter Setting at web utility.
2. Send msg.mask to this node.
   Example: msg.mask = { "MaskMatch": "1", "MaskOffset": "0", "MaskCount": "32", "MaskData": "12345678" };

### 5.3.6 Tag Kill

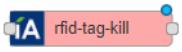
Allows a host to kill a set of tags of interest.

**Note:** A tag whose kill password value is zero will not execute a kill operation; if such a tag receives a tag-kill operation, it ignores this command.

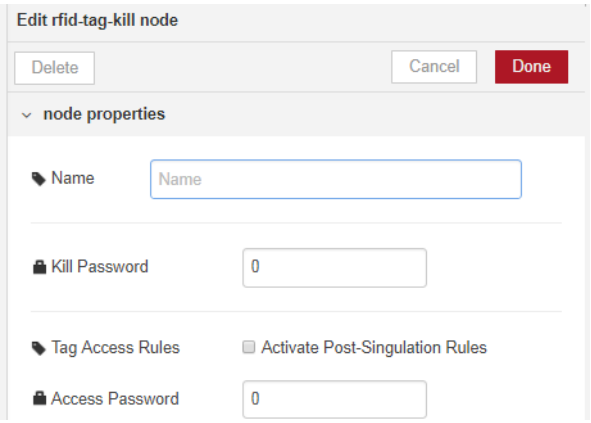
The kill password value is stored at RESERVED memory bank address 0 and address 1.

When performing tag-kill operation, the RFID reader uses only the first enabled logical antenna. (i.e. the enabled logical antenna with the smallest logical antenna port number)

1. Add the **rfid tag kill** node in Node-RED




2. Fill in the settings




- Kill Password: The kill password for the tags, and the value is expressed in hexadecimal. The range is 0x00000000~0xFFFFFFFF.
- Activate Post-Singulation Rules: Enable/disable the filter function.
- Access Password: Saves the access password for the tags. A value of zero indicates no access password.

3. Node output
   The EPC value is included in **msg.Inv.acc_data**.
   The tag access data is included in **msg.Acc.acc_data.**
   Other information please reference the appendix for detail information
4. Tag mask setting
   Enable **Activate Post-Singulation Rules** to filter the tags to be inventory.
   There are two way to set filter rule

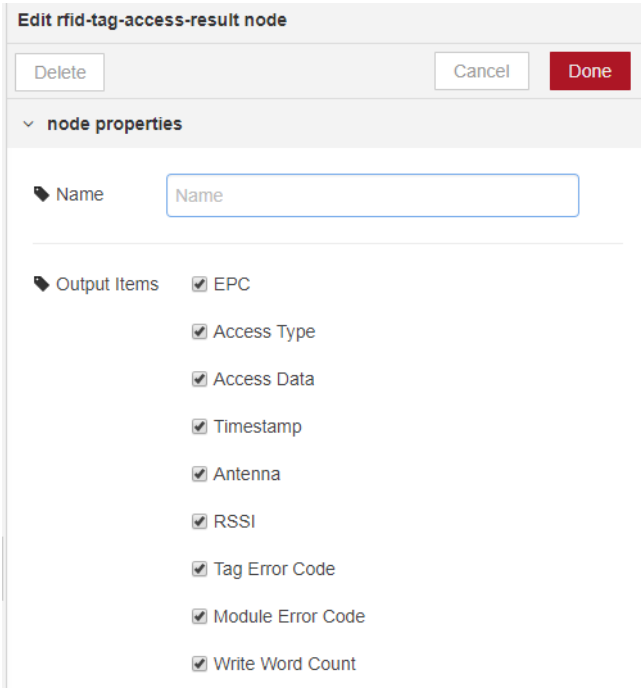
- Set Tag Filter Setting at web utility.
- Send msg.mask to this node.
  Example: msg.mask = { "MaskMatch": "1", "MaskOffset": "0", "MaskCount": "32", "MaskData": "12345678" };

### 5.3.7 Tag Access Results

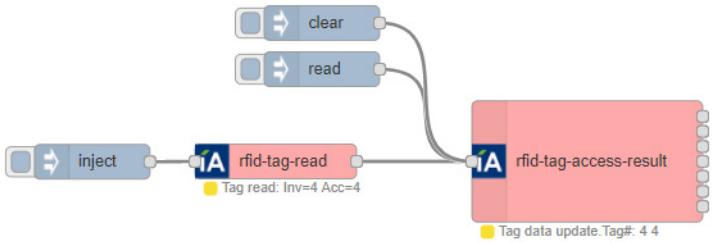
**Parsing tag access results**

1. Add the **rfid tag access result** node in Node-RED.

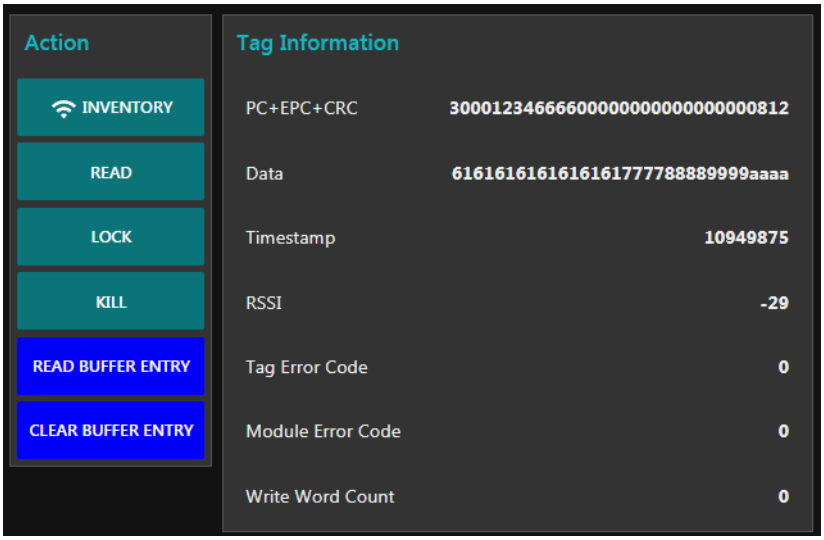

2. Fill in the settings.



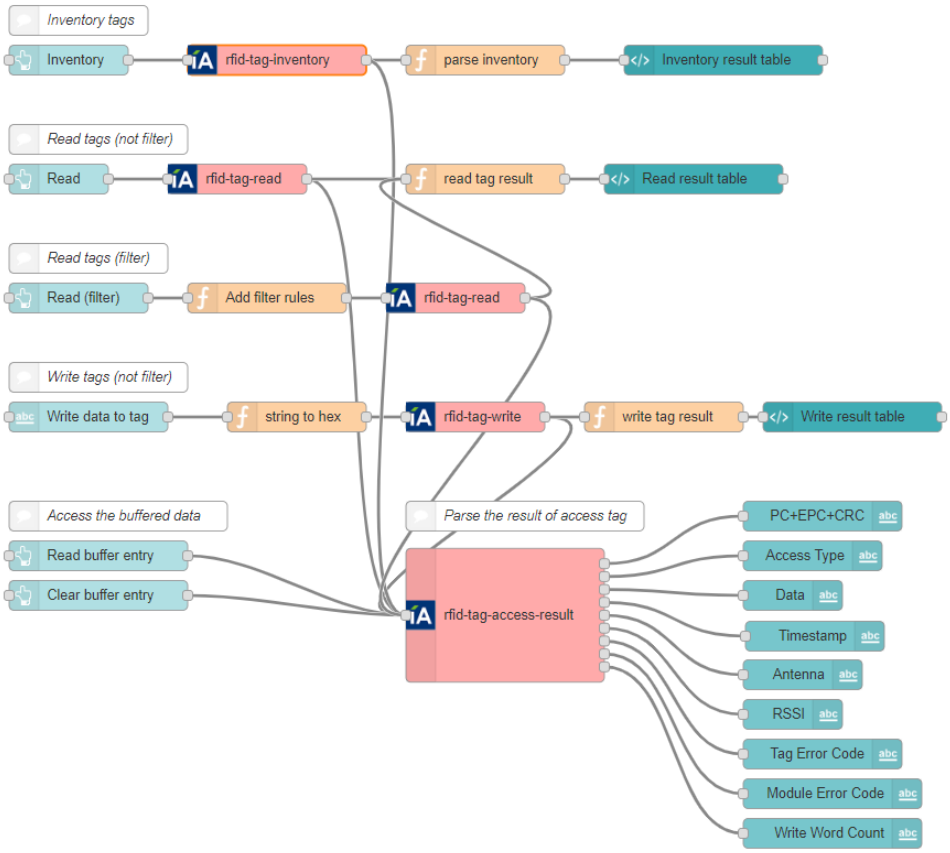
3. Use this node to parse the results of the access node.
   For example:




Tag access data are stored locally and users can get tag access information when a read Node has been added. Tag access information will be clear when a user inputs a clear node.



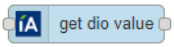
4. Please find Node-RED built-in examples.
   Import→Library→Advantech→RFID→Basic_Example




## 5.3.8 Get DIO value

**Send any input to this node to get DI/DO values for all channels**

1. Add the **get dio value** node in Node-RED.

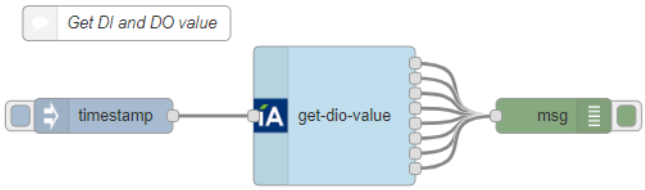



2. Fill in the settings.

- Memory Bank: EPC/TID/USER/Reserved.
- Word Offset: The offset of the first 16-bit word, zero is the first 16-bit word.
- Word Count: The number of 16-bit words to be read.
- Activate Post-Singulation Rules: Enable/disable the filter function.
- Access Password: Saves the access password for the tags. Zero value indicates no access password.

3. Node input.
   Send any input to this node to get DI/DO value of all channels.
4. Node output.
   The number of output object depends on the total number of channels.
   It then outputs msg.payload as the DI/DO status.
   It then outputs msg.error as the error status.

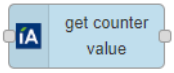
5. Please find Node-RED built-in examples
   Import→Library→Advantech→Local_IO→get_dio_values




### 5.3.9 Get counter value

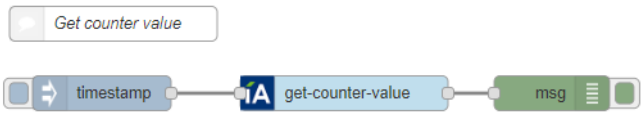
**Advantech I/O get counter value node**

1. Add the **get counter value** node in Node-RED




2. Fill in the settings
   Enter the channel number.

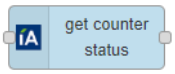

3. Node input.
   Send any input to this node to get counter values of a specific channel.
4. Node output.
   Outputs msg.payload as the counter value.
   Outputs msg.error as the error status.
5. Please find Node-RED built-in examples.
   Import→Library→Advantech→Local_IO→get_counter_values



## 5.3.10 Get counter status

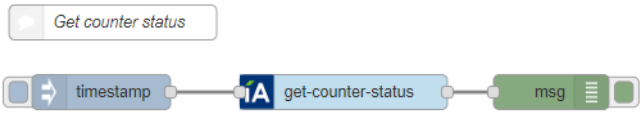
**Advantech I/O get counter status node**

1. Add the **get counter status** node in Node-RED.




2. Fill in the settings.
   Enter the channel number.

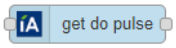

3. Node input.
   Send any input to this node to get counter status of specific channel.
4. Node output.
   Outputs msg.payload as the counter status. 0 is stop, 1 is start.
   Outputs msg.error as the error status.
5. Please find Node-RED built-in examples
   Import→Library→Advantech→Local_IO→get_counter_status




## 5.3.11 Get DO pulse count and continue mode

**Advantech I/O get do pulse node**

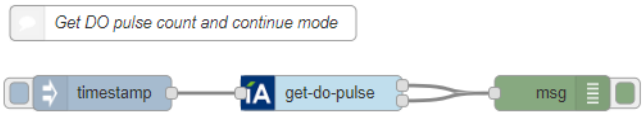
1. Add the **get do pulse** node in Node-RED.




2. Fill in the settings.
   Please fill the channel number.


3. Node input.
   Send any input to this node to get pulse status of specific channel.
4. Node output.
   Output 1
   - It then outputs msg.payload as the Pulse output count.
   - It then outputs msg.error as the error status.

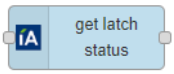
   Output 2
   - It then outputs msg.payload as the continue mode.
   - It then outputs msg.error as the error status.
5. Please find Node-RED built-in examples.
   Import→Library→Advantech→Local_IO→get_do_pulse




## 5.3.12 Get latch status

**Advantech I/O get latch status node**

1. Add the **get latch status** node in Node-RED.



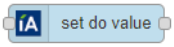

2. Fill in the settings.
   Enter the channel number.


3. Node input.
   Send any input to this node to get latch status of specific channel.
4. Node output.
   It then outputs msg.payload as the latch status.
   It then outputs msg.error as the error status.
5. Please find Node-RED built-in examples
   Import→Library→Advantech→Local_IO→get_latch_status


## 5.3.13 Set DO value

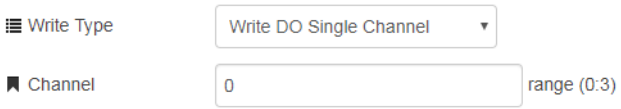
**Advantech I/O set DO value node**

1. Add the **set do value** node in Node-RED.

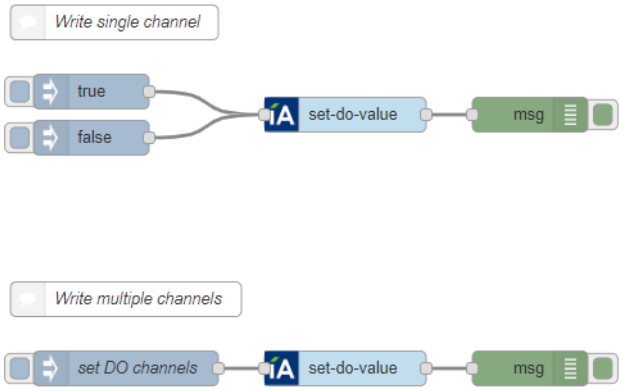



2. Fill in the settings
   Choose a write type from the drop down menu. Write type currently supported includes:

- Write DO Single Channel
- Write DO All Channels



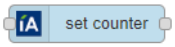
3. Node input.
   For Write DO Single Channel, msg.payload must be a number or string value of 0 or 1.
   For Write DO All Channels, msg.payload must be an array of numbers or strings with values of 0 or 1.
   Example: msg.payload = [0,0,0,0] return msg
4. Node output.
   It then outputs msg.error as the error status.
5. Please find Node-RED built-in examples
   Import→Library→Advantech→Local_IO→set_do_values




## 5.3.14 Set counter value

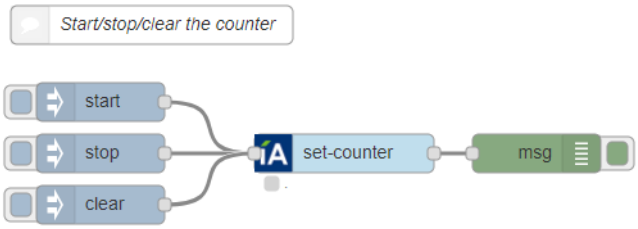
**Advantech I/O set counter value node**

1. Add the **set counter** node in Node-RED.




2. Fill in the settings.
   Enter the channel number.

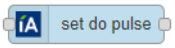

3. Node input.
   Write msg.payload to single channel.
   Send **start** string to this node to start counting.
   Send **stop** string to this node to stop counting.
   Send **clear** string to this node to clear counter value.
4. Node output.
   It then outputs msg.error as the error status.
5. Please find Node-RED built-in examples
   Import→Library→Advantech→Local_IO→set_counter




### 5.3.15 Set DO pulse

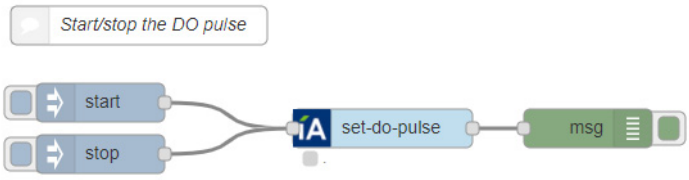
**Advantech I/O set DO pulse output node**

1. Add the **set do pulse** node in Node-RED.




2. Fill in the settings.
   Enter the channel number.
   The pulse output count range is 0~4294967295.
   If the Continue mode is enabled, the node will ignore the pulse output count.

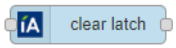

3. Node input.
   Write msg.payload to single channel.
   Send **start** string to this node to start pulse output.
   Send **stop** string to this node to stop pulse output.
4. Node output.
   It then outputs msg.error as the error status.
5. Please find Node-RED built-in examples.
   Import→Library→Advantech→Local_IO→set_do_pulse



## 5.3.16 Clear latch

**Advantech I/O set latch clear node**

1. Add the **clear latch** node in Node-RED.




2. Fill in the settings.
   Enter the channel number.


3. Node input.
   Write msg.payload to single channel.
   Send clear string to this node to clear latch.
4. Node output.
   It then outputs msg.error as the error status.
5. Please find Node-RED built-in examples
   Import→Library→Advantech→Local_IO→set_latch_clear

## 5.4 API for Development

### 5.4.1 RFID APIs

| Function | Description |
|---|---|
| unsigned char OpenDevice(char *dev,unsigned long Baud_Rate,unsigned char DataBits,unsigned char Parity,unsigned char StopBits); | Open serial port and set the related parameter to the specified serial port |
| unsigned char CloseDevice(); | Close serial port |
| unsigned long API_ConfigSetOperationMode(unsigned char r_Mode); | Set RFID antenna operation mode |
| unsigned long API_ConfigGetOperationMode(unsigned char *r_Mode); | Get RFID antenna operation mode |
| unsigned long API_AntennaPortSetState(unsigned char Port, unsigned char State); | Set RFID antenna port enable/disable status |
| unsigned long API_AntennaPortGetState(unsigned char Port, unsigned char *State, unsigned long *antennaSenseValue); | Get RFID antenna port enable/disable status |
| unsigned long API_AntennaPortSetConfiguration(unsigned char Port, AntennaPortConfig *pAntConfig); | Set RFID antenna power level, dwell time, inventory cycles, and physical port. |
| unsigned long API_AntennaPortGetConfiguration(unsigned char Port, AntennaPortConfig *pAntConfig); | Get RFID antenna power level, dwell time, inventory cycles, and physical port. |
| unsigned long API_I8K6CSetPostMatchCriteria(SingulationCriteria *pCriteria); | Setting the tag filter rule |
| unsigned long API_I8K6CGetPostMatchCriteria(SingulationCriteria *pCriteria); | Getting the tag filter rule |
| unsigned long API_I8K6CSetPostMatchMaskData(CriteriaMaskData *MaskData); | Setting the tag filter mask |
| unsigned long API_I8K6CGetPostMatchMaskData(CriteriaMaskData *MaskData); | Getting the tag filter mask |
| unsigned long API_I8K6CSetQueryTagGroup(TagGroup *pTagGroup); | Setting the tags of interest |
| unsigned long API_I8K6CGetQueryTagGroup(TagGroup *r_strcGroup); | Getting the tags of interest |
| unsigned long API_I8K6CSetTagAccessPassword(unsigned long AccessPassword); | Setting the tag access password |
| unsigned long API_I8K6CTagGetAccessPassword(unsigned long *AccessPassword); | Getting the tag access password |
| unsigned long API_I8K6CTagWriteDataBuffer(unsigned char bIndex, unsigned short wData, unsigned char bOffsetType, unsigned short wDataOffset); | Setting tag writing data buffer |
| unsigned long API_I8K6CTagReadDataBuffer(unsigned char bIndex, unsigned short *wData, unsigned short *wDataOffset); | Getting tag writing data buffer |
| unsigned long API_I8K6CTagInventory(TagAccessFlag *pTagAccessFlag, ACCESS_CALLBACK callback); | Tag inventory operation |

| | |
|---|---|
| unsigned long API_I8K6CTagRead(ReadWriteCmdParms *pReadCmdParms, TagAccessFlag *pTagAccessFlag, unsigned long accessPassword, ACCESS_CALLBACK callback); | Tag read opeartion |
| unsigned long API_I8K6CTagWrite(ReadWriteCmdParms *pWriteCmdParms, TagAccessFlag *pTagAccessFlag, unsigned long accessPassword, ACCESS_CALLBACK callback); | Tag write operation |
| unsigned long API_I8K6CTagMultipleWrite(ReadWriteCmdParms *pWriteCmdParms, TagAccessFlag *pTagAccessFlag, unsigned long accessPassword, ACCESS_CALLBACK callback); | Tag multiple write operation |
| unsigned long API_I8K6CTagKill(unsigned long killPassword, TagAccessFlag *pTagAccessFlag, unsigned long accessPassword, ACCESS_CALLBACK callback); | Tag kill operation |
| unsigned long API_I8K6CTagLock(TagPerm *lockParms, TagAccessFlag *pTagAccessFlag, unsigned long accessPassword, ACCESS_CALLBACK callback); | Tag lock operation |
| unsigned long ControlCancel(); | Canceling a tag-protocol operation |
| unsigned long ControlAbort(); | Aborting a tag-protocol operation |
| unsigned long ControlPause(); | Pausing a tag-protocol operation |
| unsigned long ControlResume(); | Resuming a tag-protocol operation |
| unsigned long API_ControlSoftReset(); | Performing a software reset |
| unsigned long API_MacGetFirmwareVersion(unsigned char *major, unsigned char *minor, unsigned char *release); | Retrieving the MAC firmware version information |
| unsigned long API_MacGetOEMCfgVersion(unsigned char *major, unsigned char *minor, unsigned char *release); | Retrieving the MAC-resident OEMCfg version information |
| unsigned long API_MacGetOEMCfgUpdateNumber(unsigned char *major, unsigned char *minor, unsigned char *release); | Retrieving the MAC-resident OEMCfg update number information |
| unsigned long API_MacClearError(); | Clearing a MAC firmware error |
| unsigned long MacGetError(unsigned char ErrorType, unsigned long *Error); | Retrieving a MAC firmware error code |
| unsigned long API_GetTemprature(unsigned char Mode, unsigned char *Temprature); | Retrieving the module temprature |
| unsigned long API_MacSetRegion(unsigned char r_Region); | Setting the region of operation |
| unsigned long API_MacGetRegion(unsigned char *r_Region,unsigned long *r_macRegionSupport); | Getting the region of operation |
| unsigned long API_TestSetFrequencyConfiguration(unsigned char r_btChannelFlag,unsigned long r_uiExactFrequecny); | Setting the test frequency configuration |
| unsigned long API_TestGetFrequencyConfiguration(unsigned char *r_btChannelFlag,unsigned long *r_uiExactFrequecny); | Getting the test frequency configuration |

Detail examples please find the WISE2800SDK'RFID

1.rfid_config.c:

This is an example to show how to control RFID antennas.

2.rfid_inventory.c:

This is an example to show how to inventory tags.

3.rfid_tag_select.c:

This is an example to show how to select the tags.

4.rfid_read_write.c:

This is an example to show how to read/write the memories of a tag.

5.rfid_lock.c:

This is an example to show how to set permissions of a tag.

6.rfid_kill.c:

This is an example to show how to kill a tag.

7.rfid_tag_algorithm.c:

This is an example to show how to set singulation algorithm and related parameter.

### 5.4.2 I/O APIs

| Function | Description |
|---|---|
| int AdamComPort_OpenComPort(char *Dev); | Open serial port |
| int AdamComPort_CloseComPort(int fd); | Close serial port |
| int AdamComPort_SetComPortState(int fd, unsigned long i_dwBaudRate, unsigned char i_byDataBits, unsigned char i_byParity, unsigned char i_byStopBits); | Set the related parameter to the specified serial port |
| unsigned long GetModuleName(int fd, char *o_szName); | Get the module name |
| unsigned long GetFirmwareVer(int fd, char *o_szVer); | Get the I/O firmware version |
| unsigned long DO_SetValue(int fd, int i_iChannel, unsigned char i_bValue); | Set the values of the specified digital output channel |
| unsigned long DO_SetValues(int fd, int i_iDOTotal, unsigned long i_dwDO); | Set the values of the digital output channels |
| unsigned long DIO_GetValues(int fd, int i_iDITotal, int i_iDOTotal, unsigned long *o_dwDI, unsigned long *o_dwDO); | Get the values of the specified digital I/O channel |
| unsigned long GetIOConfigs(int fd, int totalCh, unsigned char *o_byConfig); | Get the I/O configuration parameters |
| void ParseDOConfig(unsigned char i_byConfig, unsigned char *o_byMode); | Parse the DI configuration parameters |
| void ParseDIConfig(unsigned char i_byConfig, unsigned char *o_byMode, unsigned char *o_bRecordLastCount, unsigned char *o_bDigitalFilter, unsigned char *o_bInvert); | Parse the DO configuration parameters |
| unsigned long SetIOConfigs(int fd, int totalCh, unsigned char *i_byConfig); | Set the I/O configuration parameters |
| unsigned long GetDOConfig(int fd, int i_iChannel, unsigned char *o_byConfig); | Get the single DO configuration |
| unsigned long SetDOConfig(int fd, int i_iChannel, unsigned char i_byConfig); | Set the single DO configuration |
| unsigned long GetDIConfig(int fd, int i_iChannel, unsigned char *o_byConfig); | Get the single DI configuration |
| unsigned long SetDIConfig(int fd, int i_iChannel, unsigned char i_byConfig); | Set the single DI configuration |
| unsigned long DI_GetDiFilterMiniSignalWidth(int fd, int i_iChannel, unsigned long *o_lHigh, unsigned long *o_lLow); | Get DI filter input width |
| unsigned long DI_SetDiFilterMiniSignalWidth(int fd, int i_iChannel, unsigned long i_lHigh, unsigned long i_lLow); | Set DI filter input width |

| | |
|---|---|
| unsigned long DO_GetPulseOutputCount(int fd, int i_iChannel, unsigned char *o_bContinue, unsigned long *o_lPulseCount); | Get DO pulse output counts |
| unsigned long DO_SetPulseOutputCount(int fd, int i_iChannel, unsigned char i_bContinue, unsigned long i_lPulseCount); | Set DO pulse output counts |
| unsigned long CNT_GetValue(int fd, int i_iChannel, unsigned long *o_lValue); | Read counter or frequency value |
| unsigned long DO_GetPulseOutputWidthAndDelayTime(int fd, int i_iChannel, unsigned long *o_lPulseHighWidth, unsigned long *o_lPulseLowWidth, unsigned long *o_lHighToLowDelay, unsigned long *o_lLowToHighDelay); | Get pulse output width amd delay time |
| unsigned long DO_SetPulseOutputWidthAndDelayTime(int fd, int i_iChannel, unsigned long i_lPulseHighWidth, unsigned long i_lPulseLowWidth, unsigned long i_lHighToLowDelay, unsigned long i_lLowToHighDelay); | Set pulse output width amd delay time |
| unsigned long ALM_SetLatchClear(int fd, int i_iChannel); | Clear alarm latch |
| unsigned long CNT_GetStatus(int fd, int i_iChannel, unsigned char *o_bCounting); | Get counter start/stop status |
| unsigned long CNT_SetStatus(int fd, int i_iChannel, unsigned char i_bCounting); | Set counter start/stop status |
| unsigned long CNT_Clear(int fd, int i_iChannel); | Clear counter value |
| unsigned long DO_GetDiagnostic(int fd, int i_groupNum, unsigned char *o_sStatus); | Get DO diagnostic status |
| unsigned long SetWDTTimeout(int fd, int timeout); | Set watch dog status and timeout value |
| unsigned long GetWDTTimeout(int fd, int *timeout); | Get watch dog status and timeout value |

Detail examples please find the WISE2800SDK'IO

1.dio_example.c:

This is an example to show how to control digital I/Os.

# Appendix A

## RFID node output

**Table A.1: Inventory report**

| Name | Description | |
|---|---|---|
| pkt_header | These hex values of header information are 0x4D544949, i.e. ASCII string "MTII".<br>The fixed length of this report packet is 64 bytes. | |
| pkt_relnumber | Total relation number = variable | |
| pkt_relseq | Relation sequence number = variable | |
| rpt_ver | Report version number = 0x01 | |
| rpt_flags | Report flags: | |
| | Bit | Value and Description |
| | 0 | CRC invalid flag for backscattered tag data:<br>0 = Valid CRC<br>1 = Invalid CRC |
| | 1 | Transceiver chip:<br>0 = Indy R1000 chip<br>1 = Indy R2000 chip |
| | 2 | Serialized TID data:<br>0 = No serialized TID data in packet<br>1 = Monza TID data included (12 bytes) |
| | 3= | Extra hardware data:<br>0 = No extra hardware data in the front of inv_data<br>1 = Extra hardware data included (8 bytes) |
| | 5:4 | Reserved. Read as zero. |
| | 7:6 | Tag-data padding:<br>Number of padding bytes added to inv_data force the length of inv_data field to end on the 32-bit boundary. |
| rpt_type | Report type value = 0x0005 | |
| rpt_inflen | Information valid length = variable (greater than or equal to 3)<br>When pkt_relnumber = 1, the length of this field in words =<br>(hardware data bytes + tag data bytes + tag-data padding bytes) / 4.<br>The information data consists of hardware data, tag data and tag-data padding three parts. When pkt_relnumber = 1, each length of three parts is as follows:<br>■ The length of hardware data in bytes is 12 from byte offset 14 to 25.<br>■ The length of tag data in bytes is depending on bytes number of tag data.<br>■ The length of tag-data padding in bytes is depending on bytes number of tag-data padding of rpt_flags field.<br>For other details, see **Note 1**. | |
| rpt_seq | Increase the report sequence number progressively. | |
| ms_ctr | MTI MAC firmware millisecond counter when tag was inventoried. | |

| Table A.1: Inventory report | |
|---|---|
| nb_rssi | The narrowband receive signal strength indicator (RSSI).<br>This is the backscattered tag signal. The narrowband RSSI indication is 8-bit value. It is useful for relative signal strength indication. It is important to note that the IF LNA gain in the receive path can vary each time carrier wave is turned on, so the IF LNA gain should be taken into account. Refer to byte offsets 21:20 for a description of the ana_ctrl field, which includes the setting of the IF LNA at the time the RSSI measurement was taken.<br>Value conversion to dB formula:<br>Exponent = bits[7:3], Mantissa = bits[2:0], Mantissa_Size = 3<br>20 * log10 (2^Exponent * (1 + Mantissa / 2^Mantissa_Size))<br>Example: Value 0x48<br>Exponent = 9, Mantissa = 0<br>20 * log (2^9 * (1 + 0 / 2^3)) = 54.19 |
| wb_rssi | The wideband receive signal strength indicator (RSSI).<br>This is the backscattered tag signal. The wide-band RSSI indication is 8-bit value. It is useful for relative signal strength indication. It is important to note that the IF LNA gain in the receive path can vary each time carrier wave is turned on, so the IF LNA gain should be taken into account. Refer to byte offsets 21:20 for a description of the ana_ctrl field, which includes the setting of the IF LNA at the time the RSSI measurement was taken.<br>Value conversion to dB formula:<br>Exponent = bits[7:4], Mantissa = bits[3:0], Mantissa_Size = 4<br>20 * log10 (2^Exponent * (1 + Mantissa / 2^Mantissa_Size))<br>Example: Value 0x48<br>Exponent = 4, Mantissa = 8<br>20 * log (2^4 * (1 + 8 / 2^4)) = 27.60 |
| ana_ctrl | The value of the Indy R1000 or R2000 gain control register at time the RSSI measurement was taken - contains the IF LNA's gain info for RSSI. See the value of Transceiver chip bit of rpt_flags field for format.<br>Bits[5:4]: IF LNA gain with Indy R1000 chip (0 = 24dB, 1 = 18dB, 3 = 12dB)<br>Bits[5:3]: IF LNA gain with Indy R2000 chip (0 = 24dB, 1 = 18dB, 3 = 12dB, 7 = 6dB)<br>Other bits are reserved for future use. |
| rssi | The EPC receive signal strength indicator (RSSI). The value is the narrowband RSSI adjusted by the calibration value. The units are tenths of dBm. |
| logic_ant | The value is the current logical antenna port during the tag-singulation phase. |

**Table A.1: Inventory report**

<table>
<tr><td>acc_data</td><td>The data that was backscattered by the tag (i.e. PC + (XPC) + EPC + CRC16) during tag singulation. The data is presented in the same format as it is transmitted over the air from the tag to the RFID module - i.e. the data has not been changed to match the endianness of the host processor.<br>Tag TID data, if present, will follow the CRC16, as indicated by the Serialized TID data bits of rpt_flags field.<br>These extra hardware data, if available via command, will lead the tag data, as indicated by the Extra hardware data bit of rpt_flags field.<br>These information of extra hardware data are as follows:
<table>
<tr><td>Byte</td><td>Value and Description</td></tr>
<tr><td>0</td><td>Physical antenna port:<br>The value is the current physical antenna port during the tag-singulation phase.</td></tr>
<tr><td>1</td><td>Phase:<br>The phase data bits[6:0] represents two's complement value from -64 to 63 at the time the EPC is received. The bit 7 is reserved and always 0.<br>Value conversion to degrees formula:<br>bits[6:0] / 128 * 360<br>Value conversion to radian formula:<br>bits[6:0] / 128 * 2"π"<br>Example: Value 0x40 (= -64)<br>-64 / 128 * 360 = -180 (deg)<br>-64 / 128 * 2"π" = -3.142 (rad)<br>When the value of Transceiver chip bit of rpt_flags field = 0 (Indy R1000 chip), phase is not available and read as zero.</td></tr>
<tr><td>3:2</td><td>Temperature:<br>The value is specified in units of degree-C and a two's complement representation.</td></tr>
<tr><td>7:4</td><td>Frequency:<br>The value is specified in units of kHz.</td></tr>
</table>
</td></tr>
<tr><td>pkt_checksum</td><td>The checksum is CRC-16 calculated over the pkt_header field to the padding field.<br>Consult Section 8: Calculation of CRC-16.</td></tr>
</table>

**Table A.2: Tag access report**

| Name | Description | |
|---|---|---|
| pkt_header | These hex values of header information are 0x4D544941, i.e. ASCII string "MTIA".<br>The fixed length of this report packet is 64 bytes. | |
| pkt_relnumber | Total relation number = variable | |
| pkt_relseq | Relation sequence number = variable | |
| rpt_ver | Report version number = 0x01 | |
| rpt_flags | Report flags: | |
| | Bit | Value and Description |
| | 0 | Module access error flag:<br>0 = RFID module did not detect an error.<br>1 = RFID module detected an error. See the module_error_code field. |
| | 1 | Tag backscatter error flag:<br>0 = Tag did not backscatter an error.<br>1 = Tag backscattered an error. See the tag_error_code field. |
| | 5:2 | Reserved. Read as zero. |
| | 7:6 | Tag-data padding:<br>Number of padding bytes added to acc_data force the length of acc_data field to end on 32-bit boundary. |
| rpt_type | Report type value = 0x0006 | |
| rpt_inflen | Information valid length = variable (greater than or equal to 3)<br>When pkt_relnumber = 1, the length of this field in words = (hardware data bytes + tag data bytes + tag-data padding bytes) / 4.<br>The information data consists of hardware data, tag data and tag-data padding three parts. When pkt_relnumber = 1, each length of three parts is as follows:<br>■ The length of hardware data in bytes is 12 from byte offset 14 to 25.<br>■ The length of tag data in bytes is depending on bytes number of tag data.<br>■ The length of tag-data padding in bytes is depending on bytes number of tag-data padding of rpt_flags field.<br>For other details, see Note 1 of Section ISO 18000-6C Inventory-Response Packet. | |
| rpt_seq | Increase the report sequence number progressively. | |
| ms_ctr | MTI MAC firmware millisecond counter when tag-access operation occurred. | |

| Table A.2: Tag access report | |
|---|---|
| command | ISO 18000-6C access command:<br>0x30 - NXP ChangeConfig<br>0x31 - NXP TAM1Authenticate<br>0x32 - NXP TAM2Authenticate<br>0xC2 - Read<br>0xC3 - Write<br>0xC4 - Kill<br>0xC5 - Lock<br>0xC6 - Access<br>0xC7 - Block Write<br>0xC8 - Block Erase<br>0xC9 - Block Permalock<br>0xE0 - Untraceable |
| tag_error_code | If the tag backscattered an error (i.e. the tag backscatter error flag of rpt_flags field is set), this value is the error code that the tag backscattered. Values are:<br>0x00 - general error (catch-all for errors not covered by codes)<br>0x01 - tag does not support the specified parameters or feature<br>0x02 - insufficient privileges for the tag to perform the operation<br>0x03 - specified memory location does not exist is too small, or the tag does not support the specified EPC length<br>0x04 - tag memory location is locked or permalocked and is either not writeable or not readable<br>0x0B - tag has insufficient power to perform the memory operation<br>0x0F - tag does not support error-specific codes |
| module_error_code | If the RFID module detects an error (i.e. the module access error flag of rpt_flags field is set), and none of the error specific bits are set in the rpt_flags field, this field contains a 16-bit error code. Values are:<br>0x0000 = no error<br>0x0001 = handle mismatch<br>0x0002 = CRC error on tag response<br>0x0003 = no tag reply<br>0x0004 = invalid password<br>0x0008 = read count invalid<br>0x0009 = out of retries<br>0x000A = length mismatch<br>0xFFFF = operation failed |
| write_word_count | The number of individual words successfully written. |
| reserved | Reserved. Read as zero. |
| acc_data | If there were no errors, this field might contain the data. |
| pkt_checksum | The checksum is CRC-16 calculated over the pkt_header field to the padding field.<br>Consult Section 8: Calculation of CRC-16. |

Note:

The information data consists of hardware data, tag data and tag-data padding three parts.

When pkt_relnumber > 1, each length of three parts is as follows:

- The hardware data should be only appeared in the first packet with pkt_relseq = 1.
  The length of hardware data in bytes is 12 from byte offset 14 to 25.
- When pkt_relseq = 1, the start of tag data is byte offset 26, and the maximum length in bytes is 36 from byte offset 26 to 61.
  When pkt_relseq > 1, the start of tag data is byte offset 14, and the maximum length in bytes is 48 from byte offset 14 to 61.

- The tag-data padding is optional field which should be only appeared in the last packet with pkt_relseq = pkt_relnumber.
  The length of tag-data padding in bytes is depending on bytes number of tag-data padding of rpt_flags field.

# Appendix B

## RFID module error code

## Table B.1: Error Code Ranges/Module Table

| Error Code Number Range | Subsystem Name |
|---|---|
| 0x0000 | Command successful with no errors. |
| 0x0001 - 0x0100 | Core State Machine |
| 0x0101 - 0x0200 | Host Interface Module |
| 0x0201 - 0x0300 | RFID Protocol Modules |
| 0x0301 - 0x0400 | RFID Transceiver Control Module |
| 0x0401 - 0x0500 | GPIO, MCU support modules, OEM Config. Module |
| 0x0501 - 0x0600 | RESERVED |
| 0x0601 - 0x0700 | RFID HP-SiP Module low level interface module |
| 0x0701 - 0x0800 | BIST Module (built-in Self Test) |

## Table B.2: Error Code Details

| Code | Sub-System | Description |
|---|---|---|
| | **Core State Machine** | |
| 0x0000 | MACERR_SUCCESS | Command successful with no errors. |
| 0x0001 | CSM_ERR_UNKNOWNCMD | This error is set when an invalid command has been issued to the MAC firmware. The MAC firmware performs basic bounds checking on command values. |
| 0x0002 | CSM_ERR_PREEXECPROC | An error occurred during pre-command execution processing. This may happen if the MAC firmware is unable to transmit a Command-Begin packet to the host. |
| 0x0003 | CSM_ERR_POSTEXECPROC | An error occurred during post-command execution processing. This may happen if the MAC firmware is unable to flush the host TX buffers after the main processing of a given command is complete. |
| 0x0004 | CSM_ERR_BADENGTESTSUBCMD | This is set when an unsupported ENGTEST sub-command has been indicated via the HST_ENGTST_ARG0 register, bits 7:0.<br>FYI - BUG - currently only set if particular engineering test sub-commands have not been compiled into the MAC firmware image. Eventually this will be reported for all invalid sub-command values in HST_ENGTEST_ARG0. |
| 0x0005 | CSM_ERR_MBPRDADDR | Set if an invalid / unsupported UHF RFID transceiver register is detected in the HST_MBP_ADDR after an MBPRDREG command is issued to the MAC firmware. |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x0006 | CSM_ERR_MBPWRADDR | Set if an invalid / unsupported UHF RFID transceiver register is detected in the HST_MBP_ADDR after an MBPWRREG command is issued to the MAC firmware. |
| 0x0007 | CSM_ERR_SUBSYSINIT_CPU | Set if the CPU module fails to initialize on MAC firmware boot. |
| 0x0008 | CSM_ERR_SUBSYSINIT_DBG | Set if the Debug module fails to initialize on MAC firmware boot. |
| 0x0009 | CSM_ERR_SUBSYSINIT_CSM | Set if the Core State Machine fails to initialize on MAC firmware boot. |
| 0x000A | CSM_ERR_SUBSYSINIT_OEMCFG | Set if the OEM configuration module fails to initialize on MAC firmware boot. |
| 0x000B | CSM_ERR_SUBSYSINIT_HOSTIF | Set of the HOST interface module fails to initialize on MAC firmware boot. |
| 0x000C | CSM_ERR_SUBSYSINIT_TILIF | Set if the UHF RFID transceiver low level interface module fails to initialize on MAC firmware boot. |
| 0x000D | CSM_ERR_SUBSYSINIT_BIST | Set if the BIST module fails to initialize on MAC firmware boot. |
| 0x000F | CSM_ERR_SUBSYSINIT_GPIO | Set if the GPIO module fails to initialize on MAC firmware boot. |
| 0x0010 | CSM_ERR_SUBSYSINIT_RFTC | Set of the RF Transceiver Control module fails to initialize on MAC firmware boot. |
| 0x0011 | CSM_ERR_SUBSYSINIT_PROT | Set if the RFID Protocol module(s) fail to initialize on MAC firmware boot. |
| 0x0012 | CSM_ERR_PROTSCHED_UNKST | Set if the RFID protocol scheduler module detects an unknown state - likely indicates firmware corruption or runtime SRAM corruption by errant code. |
| 0x0013 | CSM_ERR_PROTSCHED_AMBANT | Set if the Antenna configuration dwell time and inventory round count are both zero - which is illegal and ambiguous. |
| 0x0014 | CSM_ERR_PROTSCHED_NODESC | Set if the protocol scheduler detects that no logical antennas have been enabled using the HST_ANT_DESC_CFG register bank. |
| 0x0015 | CSM_ERR_PROTSCHED_PORTDEF | Set when a bogus physical antenna port definition value is used - this likely means that the TX and RX port values are not the same - which is required for MTI RFID Development Platform. |
| 0x0016 | CSM_ERR_PROTSCHED_NOFRQCH | Set by the protocol scheduler when no frequency channels have been enabled. |
| 0x0017 | CSM_ERR_PROTSCHED_BADREGION | Set by the protocol scheduler when a bogus regulatory region has been detected in HST_REGULATORY_REGION. |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x0018 | CSM_ERR_PROTSCHED_BADFTIME | Set by the protocol schedulers FCC state machine when a bogus FCC frequency hop value has been written to HST_PROTSCH_FTIME, Bank 0 - only 100, 200, 400 milliseconds are valid values. |
| 0x0019 | CSM_ERR_PROTSCHED_FTUNETO | Not currently set by firmware. |
| 0x001A | CSM_ERR_SUBSYSINIT_OEMHWOPTS | Set if the OEM hardware-option configuration module fails to initialize on MAC firmware boot. |
| 0x001B | CSM_ERR_SUBSYSINIT_NVMEMUPD | Set if the firmware failed to initialize the NV Memory Update module at boot time. |
| 0x001C | CSM_ERR_BAD_RESET_KEY | Set if the firmware CPU module's reset device logic is called with a bogus key. This will generally only happen if the system has experienced a crash and this logic is being called through an invalid call chain - likely due to some sort of corruption. |
| 0x001D | CSM_ERR_DEV_RESET_FAILED | Set if the device reset logic fails to actually reset the device - likely due to a MCU related hardware failure or system corruption. |
| 0x001E | CSM_ERR_NVMEMUPD_ABORT_MACERRNO | Set *prior* to entering non-volatile memory update mode if the current global MAC firmware error status is indicating an error. The MAC will not enter non-volatile memory update mode if there is currently an error. The host should use the CLRERR command to clear any errors; if this doesn't work, the device may need to be manually updated using the recovery method indicated in the MAC firmware datasheet. |
| 0x001F | CSM_ERR_NVMEMUPD_INT_MEMBNDS | Set if an internal memory bounds check fails while in non-volatile memory update mode. If these errors occurred the MAC firmware tries very hard not to update non-volatile memory with bogus data. This error occurs likely due to a system corruption. |
| 0x0020 | CSM_ERR_NVMEMUPD_ENTRYKEY | Set if the non-volatile memory mode entry logic detects an invalid key. This would occur if the calling logic erroneously called the non-volatile memory logic due to system corruption / firmware error. |
| 0x0021 | CSM_ERR_NVMEMUPD_NVFLUSH | Set if, during non-volatile memory update mode, the firmware fails to write flash at the lowest level. This is likely due to flash lock bits being set (i.e. via tools like SAM-BA) or a system corruption. |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x0022 | CSM_ERR_NVMEMUPD_WRVERFAIL | Set if write verification logic fails after writing data at the lowest level to flash. This may indicate problems with the MCU device flash hardware. This can occur if the MCU device flash has been updated too many times. |
| 0x0023 | CSM_ERR_INVAL_START_CHAN | Set by the protocol scheduler if the HST_RFTC_FRQCH_CMDSTART register has been set to an invalid channel. |
| 0x0024 | CSM_ERR_PROTSCHED_UNK_ALGO | Set by the protocol scheduler if an invalid protocol algorithm has been selected via the HST_INV_CFG register. |
| 0x0025 | CSM_ERR_INVAL_PWRMODE | Set by the core state machine if an invalid power management mode has been specified in the HST_PWRMGMT register. |
| 0x0026 | CSM_ERR_PWRMODE_CORRUPT | This is set if a system corruption has occurred and the logic is unable to determine the desired power management mode. |
| 0x0027 | CSM_ERR_NVMEMUPD_TXFAIL | Set if the non-volatile memory mode logic fails to transmit a packet to the host during non-volatile memory update. |
| 0x0028 | CSM_ERR_NVMEMUPD_UPD_BOUNDS | Set during non-volatile memory update if the range indicated for updates falls outside the valid non-volatile memory ranges available on the device. |
| 0x0029 | CSM_ERR_NVMEMUPD_UNKNOWN | An unknown error has occurred during non--volatile memory updates - likely a system corruption. |
| 0x002A | CSM_ERR_NVMEMUPD_RXTO | Set during non-volatile memory mode if the firmware does not receive a packet from the host within 60 seconds. This may occur if the host has crashed or the physical interface has been removed or corrupted. |
| 0x002B | CSM_ERR_GPIO_NOTAVAIL | This error code is generated when the host / user attempts to use a GPIO pin that has previously been configured as unavailable in the OEM configuration area entry GPIO_AVAIL. |
| 0x002C | CSM_ERR_ANT_NOTAVAIL | This error code is generated when the host / user attempts to use an antenna pin that has previously been configured as unavailable in the OEM configuration area entry ANT_AVAIL. |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x002D | CSM_ERR_CMDNOTAVAILABLE | Set by the command processor when a command is invoked from the host, which has been defined, but is not available in the MAC firmware code-base. This situation can occur if, for instance, a command is disabled by means of a compile-time switch. |
| 0x002E | CSM_ERR_NOCORDICDEF | Set by the protocol scheduler when no CORDIC values are found in the OEM configuration area. CORDIC values are part of the LBT configuration. See the OEM configuration section of the firmware datasheet for more details on these settings. Cordic configuration values are only required when LBT is enabled. |
| 0x002F | CSM_ERR_SUBSYSINIT_DEBUG | Set if the firmware failed to initialize the Debug subsystem at boot time. |
| 0x0030 | CSM_ERR_SUBSYSINIT_TRACE | Set if the firmware failed to initialize the Trace subsystem at boot time. |
| 0x0031 | CSM_ERR_BUILD_TARGET_DEVICE_MISMATCH | Set if the firmware failed the Target Build and Physical Device Check at boot time. |
| 0x0032 | CSM_ERR_DIAGNOSTICS | Set if the firmware failed to properly set MAC Error diagnostic codes. Actual MAC Error may not correctly be reflected by the MAC Error register. |
| 0x0033 | CSM_ERR_SUBSYSINIT_HOSTIFREGS_INIT | Set if the MAC register default value initialization module fails to initialize on MAC firmware boot. |
| 0x0034 | CSM_ERR_SUBSYSINIT_HANDSHAKE | Set if the firmware failed to initialize the Handshake interface subsystem at boot time. |
| 0x0035 | CSM_ERR_NVMEMUPD_INVALID_MODE | Set if the HST_NV_UPDATE_CONTROL MAC register had an invalid update_mode set. |
| 0x0036 | CSM_ERR_INVALID_CMD_WHILE_IN_CRIT_ERROR | Set if a Gen2 command is attempted following a critical error during system initialization. Typically caused by a failed OEM read attempt and can usually be resolved by formatting OEM. |
| 0x0037 | CSM_ERR_CRITICAL_ERROR_UNKNOWN | Set if an unknown critical error is detected at the end of system initialization. Typically caused by a failed OEM read attempt and can usually be resolved by formatting OEM. |
| **Host Interface Module** | | |
| 0x0101 | RESERVED | RESERVED |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x0102 | HOSTIF_ERR_USBDESC | Set by the USB interface module when an unsupported descriptor TYPE has been requested by the host (i.e. not a device, string, configuration descriptor type. This may be due to compatibility problems with the USB host. |
| 0x0103 | HOSTIF_ERR_USBDESCIDX | Set by the USB interface module when an unsupported device descriptor index has been requested by the Host. |
| 0x0104 | HOSTIF_ERR_USBTXEP0 | Set by the USB interface module when it is unable to transmit the response to a request on USB endpoint 0 (aka control endpoint). This may be due to compatibility or synchronization problems with the USB host. |
| 0x0105 | RESERVED | RESERVED |
| 0x0106 | HOSTIF_ERR_USBRXBUFFSZ | Set by the USB interface module when higher level firmware requests an unsupported buffer length. This may be due to a firmware build error or corrupted firmware in flash. |
| 0x0107 | HOSTIF_ERR_RXUNKNOWN | This is set by the Host interface module when the underlying physical interface module returns an unknown error code on receive from the host. This may be due to a firmware build issue, corrupted firmware image or corrupted SRAM due to errant MAC firmware code. |
| 0x0108 | HOSTIF_ERR_TXUNKNOWN | This is set by the Host interface module when the underlying physical interface module returns an unknown error code on transmit to the Host. This may be due to a firmware build issue, corrupted firmware image or corrupted SRAM due to errant code. |
| 0x0109 | HOSTIF_ERR_BADIFSTATE | This is set when the Host interface code detects that its internal state machine out of sync. This could be due to a corrupted firmware image or corrupted SRAM due to errant MAC firmware code. |
| 0x010A | RESERVED | RESERVED |
| 0x010B | HOSTIF_ERR_REGADDR | Set by the host interface module when an invalid MAC firmware register read or write is attempted (either by the host or internally by the MAC firmware). |
| 0x010C | RESERVED | RESERVED |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x010D | HOSTIF_ERR_USBDESCINIT | This is set by the host interface module during initialization if it is unable to retrieve USB string descriptors from non-volatile memory (i.e. flash) OEM configuration area. This may be due to a corrupt or unformatted OEM configuration area. It may also be due to a firmware build issue if the OEM configuration definition is out of sync with the MAC firmware code. |
| 0x010E | HOSTIF_ERR_SELECTORBNDS | This is set when the host attempts to *write* a value to a selector type register that is out of range for that selector. |
| 0x010F | RESERVED | RESERVED. |
| 0x0110 | HOSTIF_ERR_PKTALIGN | Not currently set by MAC firmware. |
| 0x0111 | HOSTIF_ERR_BADRAWMODE | Set by the low level host interface logic if an upper level requests an unsupported raw mode. This may occur if the system is corrupted. |
| 0x0112 | HOSTIF_ERR_UNKLNKSTATE | Set by the low level host interface logic if a system corrupt occurs and the link manager cannot determine the current link state. |
| 0x0113 | HOSTIF_ERR_UNKUSBSETUP | Set by the low level host interface logic if an unknown / unsupported control command is received from the host. This may occur if the host logic and the MAC firmware logic are out of sync, in terms of the lowest level host interface (UART, USB). |
| 0x0114 | HOSTIF_ERR_UARTRXBUFFSZ | This is set if the upper layer host logic attempts to receive data and the lower layer cannot support the buffer size requested. This will happen if the system is corrupted. |
| 0x0115 | HOSTIF_ERR_RAWMODECTL | Set by the low level host interface logic if a control command is received from the host while in raw mode - which is not allowed. This would happen if the host caused the MAC firmware to enter non-volatile memory update mode, which uses the raw mode, and then the host proceeded to issue control commands. |
| 0x0116 | HOSTIF_ERR_UNKHOSTIF | Set by the host interface module at boot time if the OEM configuration area is specifying an unsupported host interface. |
| 0x0117 | HOSTIF_ERR_UNKREGSTD | Set by the host interface module at boot time if the OEM configuration area is specifying an unsupported regulatory standard. |
| 0x0118 | HOSTIF_ERR_DEBUGID | Set by host interface module if Debug Id is invalid. |
| 0x0119 | HOSTIF_ERR_DEBUGOVERFLOW | Set by host interface module if Debug Buffer overflows. |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x011A | HOSTIF_ERR_REGREADONLY | Set by the host interface module when a Read-Only MAC firmware register write is attempted by the host. |
| 0x011B | HOSTIF_ERR_REGWRITEONLY | Set by the host interface module when a Write-Only MAC firmware register read is attempted by host. |
| 0x011C | HOSTIF_ERR_BADREGIONINITVALUES | Set by the host interface module if the default region dependent parameters are invalid. |
| 0x011D | HOSTIF_ERR_INVALIDENGTESTARG | Set by an ENGTEST sub-command with an invalid argument. |
| 0x011E | HOSTIF_ERR_INVALIDSETFREQARG | Set by Set Frequency command with an invalid argument. When this error is set, the result registers will be set to 0xFFFFFFFF. |
| 0x011F | HOSTIF_ERR_INVALID_RSSI_FILTERING | Set when an invalid Inventory RSSI Filtering configuration has been configured. |
| 0x0120 | HOSTIF_ERR_INVALID_TAGACC_CNT | Set when an invalid HST_TAGACC_CNT value is specified. |
| 0x0121 | HOSTIF_ERR_INVALID_BW_MODE | Set when an invalid BlockWrite mode is specified in HST_IMPINJ_EXTENSIONS. |
| 0x0122 | HOSTIF_ERR_OEM_MAC_REG_INIT_CTRL_ERROR | Set when an invalid MAC Register Initialization pair (Control/Data) is found during the MAC Register initialization. |
| 0x0123 | HOSTIF_ERR_OEM_MAC_REG_INIT_WRITE_ERROR | Set when an invalid MAC Register Initialization write occurs found during the MAC Register initialization. |
| | **RFID Protocol Modules** | |
| 0x0200 | PROTOCOL_ERR_TRUNCATION_UNSUPPORTED | Set by protocol if truncation is set in the Select configuration register, since truncation is unsupported. |
| | **RF Transceiver Control Module** | |
| 0x0300 | RFTC_ERR_BADFRQCHAN | This is set during the PLL lock logic when a bounds check fails while checking the frequency channel configuration registers. |
| 0x0301 | RFTC_ERR_BADHOPMODE | This is set if an unsupported frequency hopping mode is detected - during the PLL lock logic. |
| 0x0302 | RFTC_ERR_PLLFAILEDTOLOCK | This is set if the PLL fails to lock. |
| 0x0303 | RFTC_ERR_XCVRADC_TIMEDOUT | This is set when the RFTC module's AUX ADC function times out waiting for an ADC conversion. |
| 0x0304 | RFTC_ERR_FILTTUNE_TIMEOUT | This is set when the RFTC module times out waiting for UHF RFID transceiver to indicate RX or TX filter tuning is complete. |
| 0x0305 | RFTC_ERR_AMBIENTTEMPTOOHOT | This is set when the RFTC module detects that the ambient temperature sensor indicates too hot. |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x0306 | RFTC_ERR_XCVRTEMPTOOHOT | This is set when the RFTC module detects that the transceiver temperature sensor indicates too hot. |
| 0x0307 | RFTC_ERR_PATEMPTOOHOT | This is set when the RFTC module detects that the PA temperature sensor indicates too hot. |
| 0x0308 | RFTC_ERR_PADELTATEMPTOOBIG | This is set when the RFTC module detects that the delta between the PA temperature and the ambient temperature is too great. |
| 0x0309 | RFTC_ERR_REVPWRLEVTOOHIGH | This is set when the reverse power level is too high as measured by the configured reverse power level threshold in the register set. |
| 0x030A | RFTC_ERR_BADIFLNAGAIN | This is set when an incorrect current gain setting is passed into the IFLNA gain adjustment logic. May indicate corrupted code. |
| 0x030B | RFTC_ERR_TXRF_BIT_FAILED | Returned by RFTC code when errors occur in transmitting a bit over the RF interface. |
| 0x030C | RFTC_ERR_TXRF_BYTE_FAILED | Returned by RFTC code when errors occur in transmitting a buffer of bytes over the RF interface. |
| 0x030D | RFTC_ERR_TXRF_EOT_FAILED | Returned by RFTC code when errors occur in transmitting an "end of transfer" command over the RF interface. |
| 0x030E | RFTC_ERR_TXRF_PREAM_FAILED | Returned by RFTC code when errors occur in transmitting a "preamble" command over the RF interface. |
| 0x030F | RFTC_ERR_TXRF_FSYNC_FAILED | Returned by RFTC code when errors occur in transmitting a "frame-sync" command over the RF interface. |
| 0x0310 | RFTC_ERR_RXRF_ISR_TIMEOUT | Indicates that the RF transceiver failed to set expected ISR bits in a timely fashion. Indicates a failure in either the RFTC state machine logic or in the RF transceiver state machine logic. |
| 0x0311 | RFTC_ERR_INVALIDLINKPARMS | This is set when invalid link parameters are detected when the filter tuning logic is run. |
| 0x0312 | RFTC_ERR_RXRF_INTERPKTTIMEOUT | This indicates a failure in either the RFTC state machine logic or in the RF transceiver state machine logic. This error can only occur if the RF transceiver starts filling its RX FIFO with received data, but fails to return the requested number of bits in a timely fashion. |
| 0x0313 | RFTC_ERR_NO_LINKPROFHDR | Not currently in use. May occur in the future when switching between link profiles if some of the required information is not properly coded in the MAC firmware. |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x0314 | RFTC_ERR_PROFILE_INVALID | This error occurs if the RF transceiver is being loaded with an invalid profile. |
| 0x0315 | RFTC_ERR_DBMVALOUTOFRANGE | Internal error. The error is the direct result of the MAC firmware having to do a "dBm to linear" conversion on a dBm measurement that is outside the range of -99dBm through +45dBm. It the unlikely event that this error is encountered, it is probably the result of a faulty RF Peak Detector, a bug in the code that computes the dBm value from the RF Peak Detector ADC reading, or a faulty external PA circuit. |
| 0x0316 | RFTC_ERR_FWDPWRLEVTOOHIGH | If, during RF power-ramping, it is determined that the RF power at the antenna port has momentarily exceeded 35dBm, or has exceeded 33dBm steady-state, this error will be thrown. Encountering this error is often the result attempting to transmit on an open antenna port or in other cases an incorrect calibration of the gross gains. Make sure an antenna is connected on the physical port in use or see MAC firmware command 0x1B for more information on how to calibrate the system. |
| 0x0317 | RFTC_ERR_NO_GROSSPWRENTRY | Internal error that may occur if memory is corrupted. |
| 0x0318 | RFTC_ERR_TARGETPWRTOOHIGH | Indicates that the target power (in MAC firmware Virtual Register 0x706) is higher than the maximum allowed output power, which is +33dBm. |
| 0x0319 | RESERVED | RESERVED. |
| 0x031A | RFTC_ERR_ANTENNADISCONNECTED | Indicates that the measured value of the antenna-sense resistor (reported in the MAC firmware Virtual Register 0x703) exceeds the threshold specified (specified in the MAC firmware Virtual register 0xB12). To determine which antenna was disconnected, the list of enabled antennas will need to be scanned for the one exceeding the threshold (this is done by iterating through all valid selectors in register 0x701 and examining the MAC_ANT_DESC_STAT register at address 0x703. |
| 0x031B | RFTC_ERR_UNREC_HWOPTFORMAT | Indicates that the OEMCFG's HW_OPTIONS_FORMAT value is not recognized by the RFTC subsystem. |
| 0x031C | RFTC_ERR_HWOPT_BADFWDPWROPT | Indicates that the forward power detection option found in OEMCFG's HW_OPTIONS0 field is not recognized by the RFTC subsystem. |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x031D | RFTC_ERR_HWOPT_BADREVPWROPT | Indicates that the reverse power detection option found in OEMCFG's HW_OPTIONS0 field is not recognized by the RFTC subsystem. |
| 0x031E | RFTC_ERR_HWOPT_BADDRMFILTOPT | Indicates that the DRM Filter option found in OEMCFG's HW_OPTIONS0 field is not recognized by the RFTC subsystem. |
| 0x031F | RFTC_ERR_HWOPT_BADAMBTEMPOPT | Indicates that ambient temperature sensor option found in OEMCFG's HW_OPTIONS0 field is not recognized by the RFTC subsystem. |
| 0x0320 | RFTC_ERR_HWOPT_BADPATEMPOPT | Indicates that PA temperature sensor option found in OEMCFG's HW_OPTIONS0 field is not recognized by the RFTC subsystem. |
| 0x0321 | RFTC_ERR_HWOPT_BADXCVRTEMPOPT | Indicates that transceiver temperature sensor option found in OEMCFG's HW_OPTIONS0 field is not recognized by the RFTC subsystem. |
| 0x0322 | RFTC_ERR_HWOPT_BADANTSENSOPT | Indicates that antenna-sense resistor sensor option found in OEMCFG's HW_OPTIONS0 field is not recognized by the RFTC subsystem. |
| 0x0323 | RFTC_ERR_BADIFLNAAGCRANGE | The range specified for the IF LNA AGC gain limits is bad. Either the "min" is higher than the "max", or the min or max setting is incorrect. |
| 0x0324 | RFTC_ERR_LPROFBADSELECTOR | When invoking the CMD_LPROF_RDXCVRREG or CMD_LPROF_WRXCVRREG commands, one of the arguments is the selector of a valid link profile. New link profile selectors cannot be created through these commands, so if a selector outside this range is passed, the RFTC_ERR_LPROFBADSELECTOR error will be generated. |
| 0x0325 | RFTC_ERR_BADXCVRADDR | One of the arguments to the CMD_LPROF_RDXCVRREG or CMD_LPROF_WRXCVRREG commands is the RF transceiver register address to configure. If the address passed is not a valid transceiver address, this error will be thrown. This error is also generated if an invalid transceiver address is detected in an OEM custom profile. |
| 0x0326 | RFTC_ERR_XCVRADDRNOTINLIST | Not all valid transceiver addresses may be configured through the link profiles. The excluded addresses include those registers which are read-only (refer to the transceiver register map) and the indirect address for the R2T command register: 0x0105. |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x0327 | RFTC_ERR_BAD_RFLNA_GAIN_REQ | Set by the RFTC module if an unsupported RFLNA gain level is requested. |
| 0x0328 | RFTC_ERR_BAD_IFLNA_GAIN_REQ | Set by the RFTC module if an unsupported IFLNA gain level is requested. |
| 0x0329 | RFTC_ERR_BAD_AGCMIX_GAIN_REQ | Set by the RFTC module if an unsupported AGC/MIXER gain level is requested. |
| 0x032A | RFTC_ERR_HWOPT_BADFWDPWRCOMPOPT | Set by the RFTC module if an unsupported compensation option is detected at OEMCFG address 0xA1. |
| 0x032B | RFTC_ERR_INVALID_PLL_DIVIDER_VALUE | This error is generated if the PLL Divider Value is zero. |
| 0x032C | RFTC_ERR_SJC_EXTERNALLOTOOLOW | This error is generated if the external LO signal level is below the threshold specified in register HST_RFTC_SJC_EXTERNALLOTHRSH. |
| 0x032D | RFTC_ERR_SJC_EXTERNALLONOTSELECTED | This error is generated if SJC is enabled, and the LO source is not external. |
| 0x032E | RFTC_ERR_BADLOSOURCE | This error is generated if the LO source is incorrectly defined in the OEM Config registers. |
| 0x032F | RFTC_ERR_GENERALRANDOMDATA | This error is generated if there is a general error in the Random Data Transmit function. |
| 0x0330 | RFTC_ERR_XVCR_HEALTH_CHECK_FAIL | This error is generated if there is transceiver health check failure and the handler is set to enable Mac Error. See OEM Config XCVR_HEALTH_CHECK_CFG. |
| 0x0331 | RFTC_ERR_INVALID_OEM_PROFILE_HEADER | This error is generated if the OEM custom profile header is invalid. |
| 0x0332 | RFTC_ERR_AUTO_READ_RX_FIFO | This error is generated if an error during the Auto Read of the Rx FIFO Read is detected. |
| 0x0333 | RFTC_ERR_DC_OFFSET_CALIBRATION | This error is general error generated if an error occurs during the DC Offset Calibration. |
| 0x0334 | RFTC_ERR_LBT_RSSI_CALIBRATION | This error is general error generated if an error occurs during the LBT RSSI Calibration. If noise floor versus calibration value do not have a significant difference this error will occur. User should check the injected reference signal for level and frequency. |
| 0x0335 | RFTC_ERR_PA_BIAS_CAL_CONFIG | This error is related to a PA Bias Calibration Configuration error. |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x0336 | RFTC_ERR_FWDPWRLEVERROR | This error is generated when the requested forward power level is not achieved during power ramp. See HST_ANT_DESC_RFPOWER for the power level requested, MAC_RFTC_PAPWRLEV for the power level achieved, and HST_RFTC_FWDPWRTHRSH for the error threshold. |
| 0x0337 | RFTC_ERR_HWOPT_BADPABIASDACCTL | Indicates that PA Bias DAC Control option found in OEMCFG's HW_OPTIONS2 field is not recognized by the RFTC subsystem. |
| 0x0338 | RFTC_ERR_PA_BIAS_CAL_MEASUREMENT | This error is related to a PA Bias Calibration measurement variation error. |
| 0x0339 | RFTC_ERR_PA_BIAS_CAL_NOT_FOUND | This error is related to a PA Bias Calibration when the target current is not found. |
| 0x033A | RFTC_ERR_GROSSGAIN_CONFIG_INVALID | This error is generated when the Gross Gain Config Value in the OEM is invalid. Min index must be less than Max, and Max must be less than the absolute max of 32. |
| 0x033B | RFTC_ERR_SJC_NOT_AVAILABLE_R500 | This error is generated if SJC is enabled with an R500 device. |
| **GPIO, MCU IO, NV Memory, OEM Configuration** | | |
| 0x0400 | IO_PERIPHERAL_PROG_ERR | This is set by the CPU module when programing IO wrong. This is likely due to errant MAC firmware code. |
| 0x0401 | IO_INVAL_RDMASK | This is set by the CPU support module when an attempt is made to read IO lines not configured for input. This may be due to internal firmware errors or the host having incorrectly configured the MTI RFID Development Platform GPIO lines. |
| 0x0402 | IO_INVAL_WRMASK | This is set by the CPU support module when an attempt is made to write IO lines not configured for output. This may be due to internal firmware errors or the host having incorrectly configured the MTI RFID Development Platform GPIO lines. |
| 0x0403 | IO_INVAL_PTR_RAM | This is set by the CPU module when a bounds check fails when accessing non-volatile memory - the caller has passed an incorrect RAM address. This is likely due to errant MAC firmware code. |
| 0x0404 | IO_INVAL_PTR_NV | This is set by the CPU module when a bounds check fails when attempting to read or write to non-volatile memory. This is likely due to errant MAC firmware code. |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x0405 | IO_INVAL_PTR_NV_ALIGN | This is set by the CPU module when a bounds check fails when attempting to read or write to non-volatile memory. This is likely due to errant MAC firmware code. |
| 0x0406 | IO_NV_LOCK_ERR | This is set by the CPU module while attempting to write to non-volatile memory (i.e. flash). This is a flash lock error and may be due to corrupted image or misconfigured firmware or hardware problems. If this error is detected by the host, it may which to attempt to read the devices OEM configuration area and save it on the host in order to preserve device specific settings. |
| 0x0407 | IO_NV_PROG_ERR | This is set by the CPU module while attempting to write to non-volatile memory (i.e. flash). This is a low-level flash write error and may be due to a misconfigured firmware image, timing problems stemming from board hardware failures, or because the flash has exceeded is limitations for writes. If this error is detected by the host, it may which to attempt to read the devices OEM configuration area and save it on the host in order to preserve device specific settings. |
| 0x0408 | IO_OEMCFG_ADDR_BOUNDS | This is set by the OEM Configuration module when an OEM configuration Address bounds check fails when accessing the OEM configuration space. This may be due to errant MAC firmware code or errant Host code. |
| 0x0409 | IO_OEMCFG_NV_BOUNDS | This is set by the OEM Configuration module when a non-volatile memory bounds check fails when accessing the OEM configuration space. This may be due to errant MAC firmware code or errant Host code. |
| 0x040A | IO_OEMCFG_FMT_KEY | This is set by the OEM Configuration module's format facility used as the code calling it fails to pass in the correct "format key" argument. This is a failsafe to prevent errant code from inadvertently reformatting flash - due to an invalid branch instruction, etc. This will occur when errant code jumps to the format facility incorrectly. |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x040B | IO_OEMCFG_FLUSH | This is set by the OEM Configuration module when it fails to flush in memory buffers to non-volatile memory. This may be due to a misconfigured firmware image, timing problems stemming from board hardware failures, or because the flash has exceeded its limitations for writes. If this error is detected by the host, it may switch to attempt to read the device's OEM configuration area and save it on the host in order to preserve device specific settings. |
| 0x040C | IO_OEMCFG_FORMAT | This is set by the OEM Configuration module when it fails to detect the correct low level file system headers for the OEM configuration area. This means that the OEM configuration area has not been formatted - due to a misconfigured board or that the OEM Configuration area has become corrupt and should not be trusted without attempting recovery or reconfiguration. |
| 0x040D | IO_INVAL_IORSVD | This is set by the CPU module when an attempt is made to configure reserved IO pins. This is likely due to a misconfigured firmware build or errant MAC firmware code. |
| 0x040E | IO_OEMCFG_STRING_TYPE | This is set by the OEM Configuration module when an invalid string type is selected. |
| 0x040F | IO_OEMCFG_STRING_LENGTH | This is set by the OEM Configuration module when an invalid string length is entered. |
| 0x0410 | IO_OEMCFG_STRING_CHARACTER | This is set by the OEM Configuration module when an invalid character is entered. |
| 0x0411 | IO_OEMCFG_STRING_CURRENT_INVALID | This is set by the OEM Configuration module when a string read cannot be read correctly since the current string has an invalid header. |
| 0x0412 | IO_OEMCFG_FORMAT_KEY_INVALID | This is set by the OEM Configuration module when the generated key does not match the check key when attempting to format the OEM Configuration space. |
| 0x0413 | IO_OEMCFG_FORMAT_CONFIGURATION_INVALID | This is set by the OEM Configuration module when an invalid format configuration is specified. |
| 0x0414 | IO_INVAL_NV_SECTOR | This is set by the CPU module while attempting to lock or unlock a flash sector and the specified sector is invalid. |
| **Low Level RFID HP-SiP Module Interface** | | |

| Table B.2: Error Code Details | | |
|---|---|---|
| 0x0601 | TILDENIF_ERR_ADDRMISMAT | This is set by the UHF RFID transceiver interface module when an UHF RFID transceiver register read, when configured for Serial port mode, returns the incorrect register address in the serial response frame. This could be due to board or UHF RFID transceiver hardware problems or errant MAC firmware code. |
| 0x0602 | TILDENIF_ERR_RDFAILSAFE | This is set by the UHF RFID transceiver interface module when failsafe logic is activated due to no response from the UHF RFID transceiver. This happens on UHF RFID transceiver register reads. This could be due to board or UHF RFID transceiver hardware problems. |
| 0x0603 | TILDENIF_ERR_INVALPWRST | Set by the low level interface logic if, during power management, an invalid power state is requested. This will likely only occur if the system is corrupt. |
| 0x0604 | TILDENIF_ERR_INVALID_SETTING_R500 | Set by the low level interface logic if, during a write, an invalid setting is selected. |
| | **Built-In Self Test** | |
| 0x0701 | BIST_ERR_RF_IO_REG_CHK | This error code is set during firmware boot when the Built-In Self Test code is executed. This error indicates that certain register power up defaults on UHF RFID transceiver were not detected - possibly indicating a hardware problem. |
| 0x0702 | BIST_ERR_RF_REG_BITS | This error code is set during firmware boot when the Built In Self Test code is executed. This error indicates that a walking 1's or walking 0's bus test failed - possibly indicating a hardware problem. |

# Appendix C

## RFID Frequency Channel Tables

## C.1 United States/Canada/Mexico Region Frequency Channel Table

The frequency range of those regions, which are United States, Canada and Mexico regions, is from 902 to 928 MHz. A table of all 50 channels is shown in Table C.1.

**Table C.1: Frequency Channel Table of US Band**

| Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 902.75 | 2 | 903.25 | 3 | 903.75 | 4 | 904.25 | 5 | 904.75 |
| 6 | 905.25 | 7 | 905.75 | 8 | 906.25 | 9 | 906.75 | 10 | 907.25 |
| 11 | 907.75 | 12 | 908.25 | 13 | 908.75 | 14 | 909.25 | 15 | 909.75 |
| 16 | 910.25 | 17 | 910.75 | 18 | 911.25 | 19 | 911.75 | 20 | 912.25 |
| 21 | 912.75 | 22 | 913.25 | 23 | 913.75 | 24 | 914.25 | 25 | 914.75 |
| 26 | 915.25 | 27 | 915.75 | 28 | 916.25 | 29 | 916.75 | 30 | 917.25 |
| 31 | 917.75 | 32 | 918.25 | 33 | 918.75 | 34 | 919.25 | 35 | 919.75 |
| 36 | 920.25 | 37 | 920.75 | 38 | 921.25 | 39 | 921.75 | 40 | 922.25 |
| 41 | 922.75 | 42 | 923.25 | 43 | 923.75 | 44 | 924.25 | 45 | 924.75 |
| 46 | 925.25 | 47 | 925.75 | 48 | 926.25 | 49 | 926.75 | 50 | 927.25 |

## C.2 Europe Region Frequency Channel Table (ETSI EN 302 208)

The frequency range of Europe region is from 865.6 to 867.6 MHz. A table of all 4 channels is shown in Table C.2.

**Table C.2: Frequency Channel Table of EU Band**

| Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) |
|---|---|---|---|---|---|---|---|
| 1 | 865.7 | 2 | 866.3 | 3 | 866.9 | 4 | 867.5 |

## C.3 Europe2 Region Frequency Channel Table(ETSI EN 300 220)

The frequency of Europe2 region is only 869.85 MHz. A table of 1 channel is shown in Table C.3.

**Table C.3: Frequency Channel Table of EU2 Band**

| Channel | Frequency (MHz) |
|---|---|
| 1 | 869.85 |

## C.4 Taiwan Region Frequency Channel Table

The frequency range of Taiwan region is from 922 to 928 MHz. A table of all 12 channels is shown in Table C.4.

**Table C.4: Frequency Channel Table of TW Band**

| Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 922.25 | 2 | 922.75 | 3 | 923.25 | 4 | 923.75 | 5 | 924.25 |
| 6 | 924.75 | 7 | 925.25 | 8 | 925.75 | 9 | 926.25 | 10 | 926.75 |
| 11 | 927.25 | 12 | 927.75 | | | | | | |

## C.5 China Region Frequency Channel Table

The frequency range of China region is from 920.5 to 924.5 MHz. A table of all 16 channels is shown in Table C.5.

**Table C.5: Frequency Channel Table of CN Band**

| Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 920.625 | 2 | 920.875 | 3 | 921.125 | 4 | 921.375 | 5 | 921.625 |
| 6 | 921.875 | 7 | 922.125 | 8 | 922.375 | 9 | 922.625 | 10 | 922.875 |
| 11 | 923.125 | 12 | 923.375 | 13 | 923.625 | 14 | 923.875 | 15 | 924.125 |
| 16 | 924.375 | | | | | | | | |

## C.6 South Korea Region Frequency Channel Table

The frequency range of South Korea is from 917 to 920.8 MHz. A table of all 6 channels is shown in Table C.6.

**Table C.6: Frequency Channel Table of KR Band**

| Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 917.3 | 2 | 917.9 | 3 | 918.5 | 4 | 919.1 | 5 | 919.7 |
| 6 | 920.3 | | | | | | | | |

## C.7 Australia/New Zealand Region Frequency Channel Table

The frequency range of both Australia and New Zealand regions is from 920 to 926 MHz. A table of all 7 channels is shown in Table C.7.

**Table C.7: Frequency Channel Table of AU/NZ Band**

| Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 922.25 | 2 | 922.75 | 3 | 923.25 | 4 | 923.75 | 5 | 924.25 |
| 6 | 924.75 | 7 | 925.25 | | | | | | |

## C.8 Brazil Region Frequency Channel Table

The frequency range of Brazil region is from 902 to 907.5 MHz and from 915 to 928 MHz. A table of all 35 channels is shown in Table C.8.

**Table C.8: Frequency Channel Table of BR Band**

| Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 902.75 | 2 | 903.25 | 3 | 903.75 | 4 | 904.25 | 5 | 904.75 |
| 6 | 905.25 | 7 | 905.75 | 8 | 906.25 | 9 | 906.75 | 10 | 907.25 |
| 11 | 915.25 | 12 | 915.75 | 13 | 916.25 | 14 | 916.75 | 15 | 917.25 |
| 16 | 917.75 | 17 | 918.25 | 18 | 918.75 | 19 | 919.25 | 20 | 919.75 |
| 21 | 920.25 | 22 | 920.75 | 23 | 921.25 | 24 | 921.75 | 25 | 922.25 |
| 26 | 922.75 | 27 | 923.25 | 28 | 923.75 | 29 | 924.25 | 30 | 924.75 |
| 31 | 925.25 | 32 | 925.75 | 33 | 926.25 | 34 | 926.75 | 35 | 927.25 |

## C.9 Israel Region Frequency Channel Table

The frequency range of Israel region is from 915 to 917 MHz. A table of all 2 channels is shown in Table C.9.

**Table C.9: Frequency Channel Table of IL Band**

| Channel | Frequency (MHz) | Channel | Frequency (MHz) |
|---|---|---|---|
| 1 | 915.75 | 2 | 916.25 |

## C.10 India Region Frequency Channel Table

The frequency range of India region is from 865 to 867 MHz. A table of all 2 channels is shown in Table C.10.

**Table C.10: Frequency Channel Table of IN Band**

| Channel | Frequency (MHz) | Channel | Frequency (MHz) |
|---|---|---|---|
| 1 | 865.7 | 2 | 866.3 |

## C.11 Japan Region Frequency Channel Table

The frequency range of Japan region is from 916.7 to 920.9 MHz. A table of all 4 channels is shown in Table C.11.

**Table C.11: Frequency Channel Table of JP Band**

| Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) |
|---|---|---|---|---|---|---|---|
| 1 | 916.8 | 2 | 918.0 | 3 | 919.2 | 4 | 920.4 |

## C.12 Japan2 Region Frequency Channel Table (with LBT)

The frequency range of Japan2 region is from 916.7 to 920.9 MHz. A table of all 6 channels is shown in Table C.12.

**Table C.12: Frequency Channel Table of JP2 Band**

| Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) | Channel | Frequency (MHz) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 916.8 | 2 | 918.0 | 3 | 919.2 | 4 | 920.4 | 5 | 920.6 |
| 6 | 920.8 | | | | | | | | |

ADVANTECH

*Enabling an Intelligent Planet*

## www.advantech.com

**Please verify specifications before quoting. This guide is intended for reference purposes only.**

**All product specifications are subject to change without notice.**