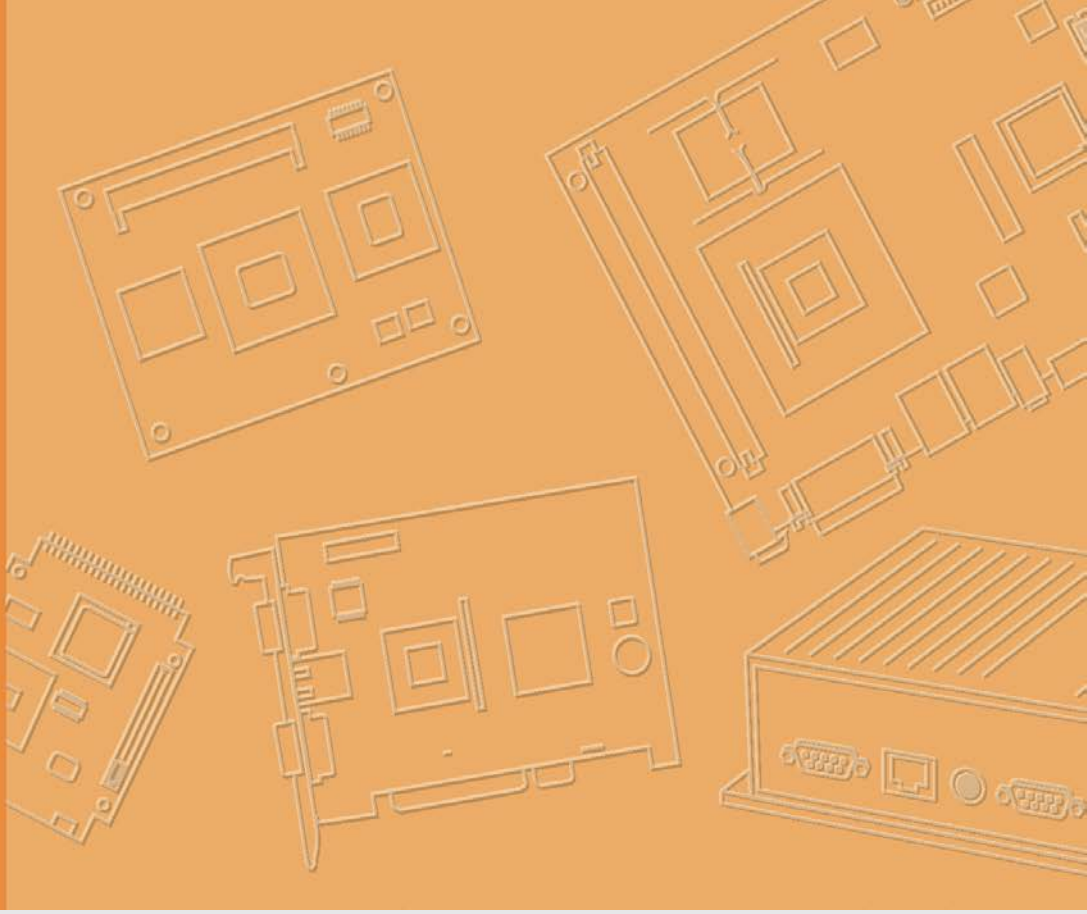


User Manual

TREK-120

Cold Chain Sensor, Gateway & NFC reader

ADVANTECH

Enabling an Intelligent Planet

Copyright

The documentation and the software included with this product are copyrighted 2019 by Advantech Co., Ltd. All rights are reserved. Advantech Co., Ltd. reserves the right to make improvements in the products described in this manual at any time without notice. No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission of Advantech Co., Ltd. Information provided in this manual is intended to be accurate and reliable. However, Advantech Co., Ltd. assumes no responsibility for its use, nor for any infringements of the rights of third parties, which may result from its use.

Acknowledgements

Windows is registered trademarks of Microsoft.

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

Product Warranty (2 years)

Advantech warrants to you, the original purchaser, that each of its products will be free from defects in materials and workmanship for two years from the date of purchase.

This warranty does not apply to any products which have been repaired or altered by persons other than repair personnel authorized by Advantech, or which have been subject to misuse, abuse, accident or improper installation. Advantech assumes no liability under the terms of this warranty as a consequence of such events.

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

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

1. Collect all the information about the problem encountered. (For example, CPU speed, Advantech products used, other hardware and software used, etc.) Note anything abnormal and list any onscreen messages you get when the problem occurs.
2. Call your dealer and describe the problem. Please have your manual, product, and any helpful information readily available.
3. If your product is diagnosed as defective, obtain an RMA (return 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.

Part No. Edition 1

Jan 2019

Declaration of Conformity

CE

This product has passed the CE test for environmental specifications. Test conditions for passing included the equipment being operated within an industrial enclosure. In order to protect the product from being damaged by ESD (Electrostatic Discharge) and EMI leakage, we strongly recommend the use of CE-compliant industrial enclosure products.

FCC Class B

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

Reorient or relocate the receiving antenna

Increase the separation between the equipment and receiver.

Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

Consult the dealer or an experienced radio/TV technician for help.

This device complies with Part 15 FCC Rules.

Operation is subject to the following two conditions.

(1) This device may not cause harmful interference , and

(2) The device must accept any interference received, including interference may cause undesired operation.

FCC Caution :

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

FCC RF Radiation Exposure Statement :

This device meets the government`s requirements for exposure to radio waves. This device is designed and manufactured not to exceed the emission limits for exposure to radio frequency (RF) energy set by the Federal Communications Commission of the U.S. Government.

This device complies with FCC radiation exposure limits set forth for an uncontrolled environment. In order to avoid the possibility of exceeding the FCC radio frequency exposure limits, human proximity to the antenna shall not be less than 20cm (8 inches) during normal operation.

Changes or modifications not expressly approved by the party responsible for compliance could void the user`s authority to operate the equipment.

Technical Support and Assistance

1. Visit the Advantech web site at http://support.advantech.com 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

Warnings, Cautions and Notes

Warning! *Warnings indicate conditions, which if not observed, can cause personal injury!*

Caution! *Cautions are included to help you avoid damaging hardware or losing data. e.g.*

There is a danger of a new battery exploding if it is incorrectly installed. Do not attempt to recharge, force open, or heat the battery. Replace the battery only with the same or equivalent type recommended by the manufacturer. Discard used batteries according to the manufacturer's instructions.

Note! *Notes provide optional additional information.*

Document Feedback

To assist us in making improvements to this manual, we would welcome comments and constructive criticism. Please send all such - in writing to: support@advantech.com

Packing List

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

Description	Q`ty
TREK-120	1
Mounting screws	2

Ordering Information

Part Number	Description
TREK-120-ANF000A00	Temperature/humidity/G sensor by NFC
TREK-120-ANLNA0A00	Temperature/humidity/G sensor by NFC+LoRa NA902/TW922
TREK-120-ANLEU0A00	Temperature/humidity/G sensor by NFC+LoRa EU868
TREK-120-ANLCN0A00	Temperature/humidity/G sensor by NFC+LoRa CN470
TREK-120-ANR000A00	Cold Chain NFC reader
TREK-120-ALG000A00	LoRa gateway dongle 525MHz ~ 928MHz
TREK-120-ALG000B00	LoRa gateway dongle 433MHz ~ 510MHz

Safety Instructions

1. Read these safety instructions carefully.
2. Keep this User Manual for later reference.
3. Keep this equipment away from fire.
6. Do not leave this equipment in an environment unconditioned where the storage temperature under -40° C (-40° F) or above 85° C (185° F), it may damage the equipment. Operating temperature: -30°C~70°C.
8. Do not operate this equipment in an environment temperature may over 70°C(149° F). The surface temperature of plastic chassis may be hot.
9. All cautions and warnings on the equipment should be noted.
12. If the equipment is not used for a long time, stop recording by setting utility.
13. Never pour any liquid into an opening. This may cause fire or electrical shock.
14. Never open the equipment. For safety reasons, the equipment should be opened only by qualified service personnel.
15. 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.
16. CAUTION: The computer is provided with a battery-powered real-time clock cir- cuit. There is a danger of explosion if battery is incorrectly replaced. Replace only with same or equivalent type recommended by the manufacture. Discard used batteries according to the manufacturers instructions.
17. This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions:

    (1) this device may not cause harmful interference, and

    (2) this device must accept any interference received, including interference that may cause undesired operation.
18. CAUTION: Always completely disconnect the power cord from your chassis whenever you work with the hardware. Do not make connections while the power is on. Sensitive electronic components can be damaged by sudden power surges.
19. CAUTION: Always ground yourself to remove any static charge before touching the

motherboard, backplane, or add-on cards. Modern electronic devices are very sensitive to static electric charges. As a safety precaution, use a grounding wrist strap at all times. Place all electronic components on a static-dissipative surface or in a static-shielded bag when they are not in the chassis.

20. CAUTION: Any unverified component could cause unexpected damage. To ensure the correct installation, please always use the components (ex. screws) provided with the accessory box.

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 mainboard or other cards while the system 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.

This product is intended to be supplied by an AA wide temperature battery, if need further assistance with purchasing the battery, please contact Advantech for further information.

Chapter 1

General Information

This chapter gives background information on the TREK-120 Cold chain Sensor

Sections include:

- **Introduction**
- **General Specifications**
- **Dimensions**

1.1 Introduction

TREK-120 LoRa temperature and humidity sensors aimed at cold chain logistics and supermarket applications. In response to industry demands for rugged cold chain fleet management solutions, Advantech developed its TREK-120 wireless temperature and humidity sensors that can be integrated with an NFC reader and LoRa gateway for wireless data collection and transmissions. Functioning like a data logger, TREK-120 sensors are designed to collect and log all temperature and humidity data accurately and precisely without data losses. The collected data can then be downloaded using an NFC reader (or any handheld device with NFC capabilities) or wirelessly transmitted to the cloud via a LoRa gateway to enable real-time monitoring. The inclusion of LoRa technology not only enables long-distance (up to 50 meters) wireless transmissions with excellent signal penetration, but also reduces the sensor's overall power consumption. However, even with the use of NFC and LoRa technology, the sensor battery supports at least 1 year of operation (depending on operating environment) and can be easily replaced for convenient management.

To withstand operation in refrigerated vehicles and cold chain logistics fleets, the TREK-120 sensors are IP65 rated for protection from water and dust ingress and support a wide operating temperature range (-20 ~ 65 °C/-4 ~ 149 °F). Moreover, the sensor design is compact, lightweight (108 g), and cableless, and can be easily installed using magnets or an adhesive backing, eliminating the need for additional cables and invasive vehicle modifications. The TREK-120 sensors are also equipped with a drop detection sensor that facilitates product tracking and traceability monitoring. Furthermore, in terms of development potential, Advantech also provides specific software APIs and SDKs that enable customization and cloud-based applications for centralized fleet management.

For operation in diverse usage environments, TREK-120 sensors can be paired with a LoRa gateway or NFC reader. Additionally, TREK-120 sensors support management using mobile devices that feature Windows or Android operating systems to enable uninterrupted cold chain management. Finally, to facilitate development, Advantech offers software APIs and SDKs (for Windows/Android) that allow system integrators to develop unique applications and cold chain application software aimed at optimizing vehicle diagnostics and facilitating cloud-based centralized management.

1.2 General Specifications

TREK-120 Specifications

Temperature	Measurement Range	-30 ~ 70 °C (-22 ~ 158 °F)
	Accuracy Range	±0.5 °C from 0 ~ 20 °C(±32.9 °F from 32 ~ 68 °F) ±1 °C from -30 ~ 70 °C (±33.8 °F from -22 ~ 158 °F)
	Resolution	0.1 °C (32.18 °F)
Frequency	NFC	13.56MHz
	LoRa	863-870MHz for Europe 920-925MHz for North America/Taiwan 470-510 MHz for China
Humidity	Measurement Range	0 ~ 100% RH
	Accuracy Range	±3% from 0 to 80%
3-Axis accelerometer		ST IIS2DH with free fall detection. Default value is 350mg
Data Storage Capacity		65Kbits for 2000 temperature and humidity values + 30 free fall values

LED Indicators		1 x Power status 1 x Alarm
Buttons		1x Start button
Battery		3.6V/2000mAh wide temperature battery (-30 ~ 70 °C)
Data Transmissions		NFC only and NFC + LoRa
Mechanical	Mount Options	2x Screw holes for mounting. Back sticker 2x Magnet
	Dimensions (W x D x H)	123.97 x 65.21 x 23.14 mm (4.88 x 2.56 x 0.91 mm)
	Weight	108g (0.23Ib)
Environment	Operating Temperature	-30 ~ 70 °C (-22 ~ 158 °F)
	Storage Temperature	-40 ~ 85 °C (-40 ~ 185 °F)
	IP Rating	IP65
	Drop Tolerance	4 ft. drop onto concrete
	Certifications	CE/FCC/NCC

LoRa Gateway Dongle

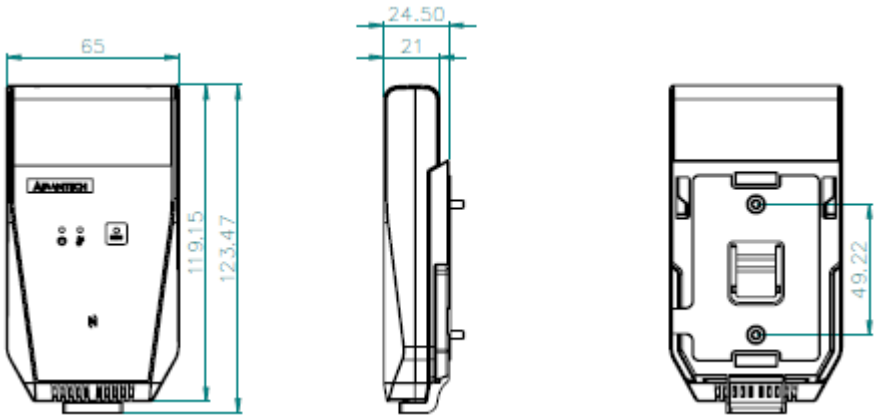
- Low-power middle range receiver operating from 433MHz to 928MHzfrequency Bands
- Host interface: USB
- Enhanced noise filtering for better RF performance
- Easy connection to Flex antenna
- Dimension: 82x62x20 mm, weight: 100g

NFC Reader

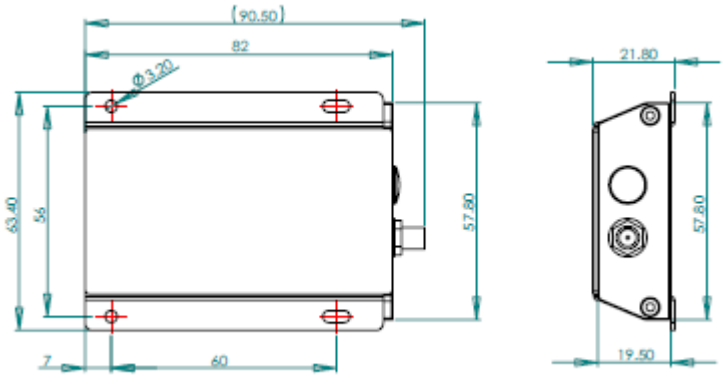
RF Operating Frequency	13.56MHz +/- 7kHz (nominal)
RF Output Power	200 mW
RF Output Impendence	50 Ohm
Host Interface	Micro USB
Input Voltage	DC 5V
Operating Temperature	0 ~ 70°C (32 ~ 158°F)
Storage Temperature	-25 ~ 85°C (13 ~ 185°F)
Dimensions	121.63 x 74.18 x 15.6 mm (4.79 x 2.92 x 0.61 in)
Weight	52g (0.11 Ib)
Certifications	CE/FCC/NCC

1.3 Dimensions

TREK-120 Cold Chain Sensor



LoRa gateway dongle



NFC reader

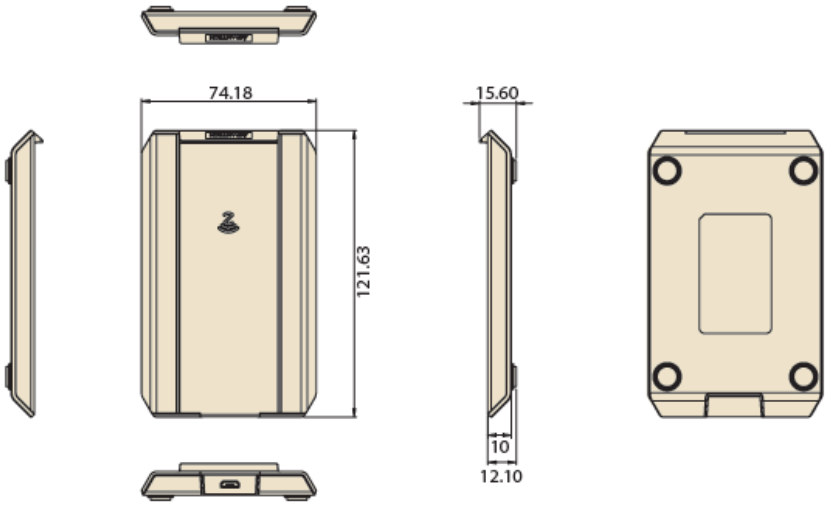


Figure 1.1 TREK-120 dimensions

Chapter 2

System Setup

This chapter details system setup on TREK-120

Sections include:

- A Quick Tour of TREK-120
- Installation Procedures

2.1 A Quick Tour of TREK-120

Before starting to set up TREK-120, take a moment to become familiar with the locations and functions of the buttons and indicators, which are illustrated in the figures below.

2.1.1 Outlook

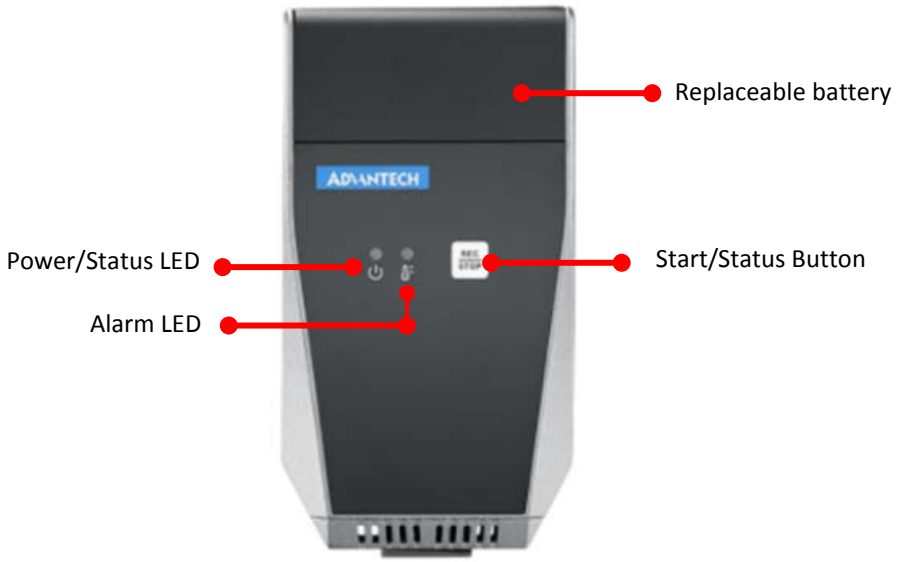


Figure 2.1 Front view of TREK-120

2.1.2 LED behavior

Logger狀態	PWR LED	Thermal LED
Stop/ Non-set 關閉中未配置	**Normal**: Green :Battery, sensor status ok.	NA
	Error: Red : Battery Low, sensors no response/fault, data full.	
	1s on, 1s off, 2time.	
Stop/ Set 關閉中已配置 (Button 啟動 Logger)	**Normal**: Green: Start logging.	**Normal**: Green, Temp, Humidity, G-sensor
	Error: Red : Battery Low, sensors no response , data full if can`t rewrite.	**Error**: Red, **Over criteria.(each condition)**
	0.5s on, 05.s off, 2time.	0.5s on, 05.s off, 2time.
Start 使用中	**Normal**: Green, Start logging.	Normal: Green, Temp, **Humidity, G-sensor,**
	Error: Red, Battery Low, sensors no response , data full if can`t rewrite..	**Error**: Red, Over criteria. (each condition)
	0.5s on, 05.s off, 2time.	0.5s on, 05.s off, 2time.

Figure 2.2 LED behavior of TREK-120

2.1.3 Mounting Way

Figure 2.3 Mounting way of TREK-120

2.1.4 Replace Battery

The procedure of replacing battery

- Release 2 screws on battery cover
- Open battery cover
- Take off the battery
- Press “Start“ button. The sensor Power LED shows “Red”
- Install the new battery
- Press “Start“ button. The sensor Power LED shows “Green”

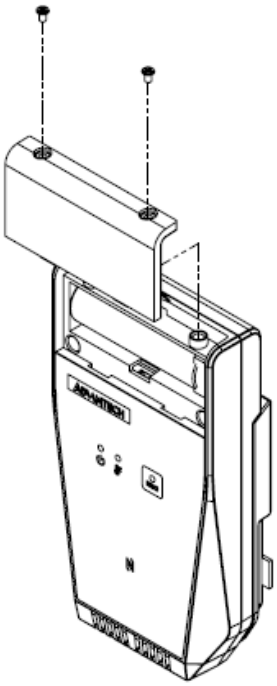
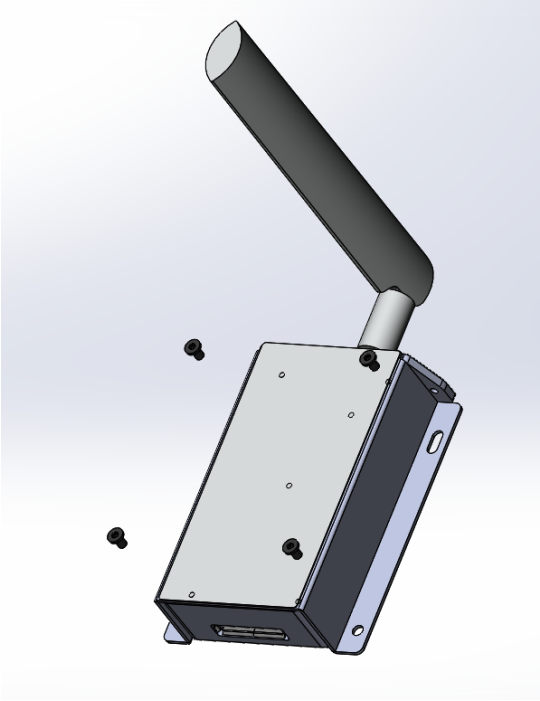


Figure 2.4 Replace battery of TREK-120

2.1.5 Mounting of LoRa gateway dongle

There`s a mounting kit on the LoRa gateway dongle. Suggest using 4pcs M3 or M2.5 self-tapping screw or machine screw to fix the gateway dongle.



2.2 Installation Procedures

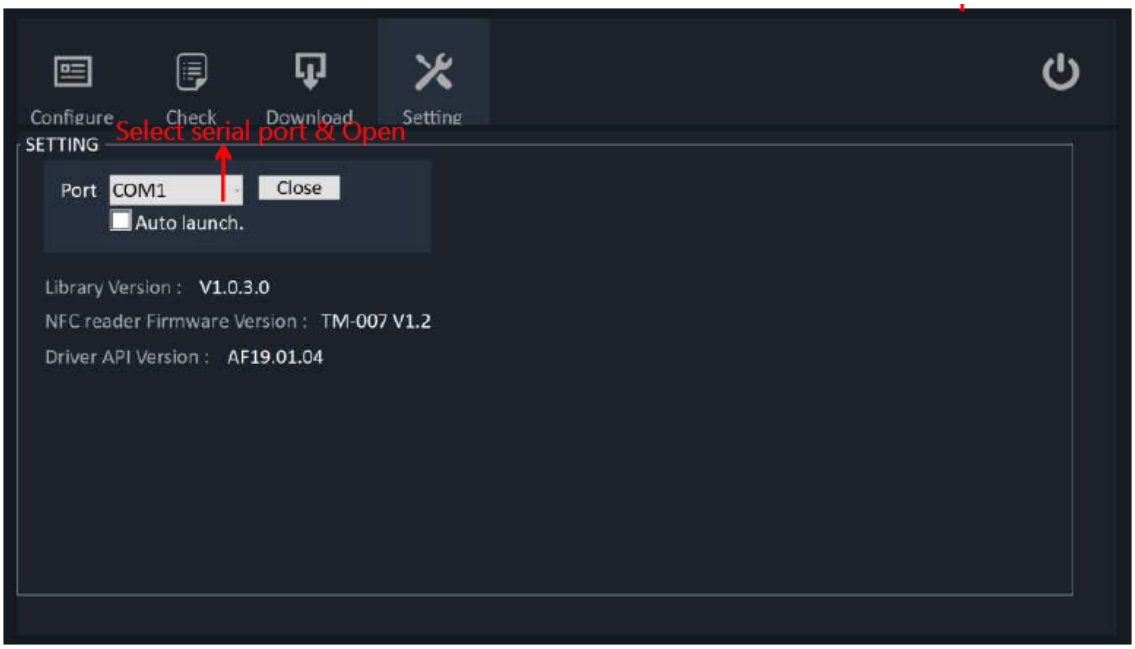
2.2.1 Setup TREK-120 NFC reader

a. Connect NFC reader to a PC or notebook USB port.

b. Download Cold Chain SDK from the website :

c. Unzip SDK file (Password is the same as the file name) and Launch coldchain_demo.exe.

d. Go to “Setting”, and then select COM port which NFC reader connected then press “Open”.



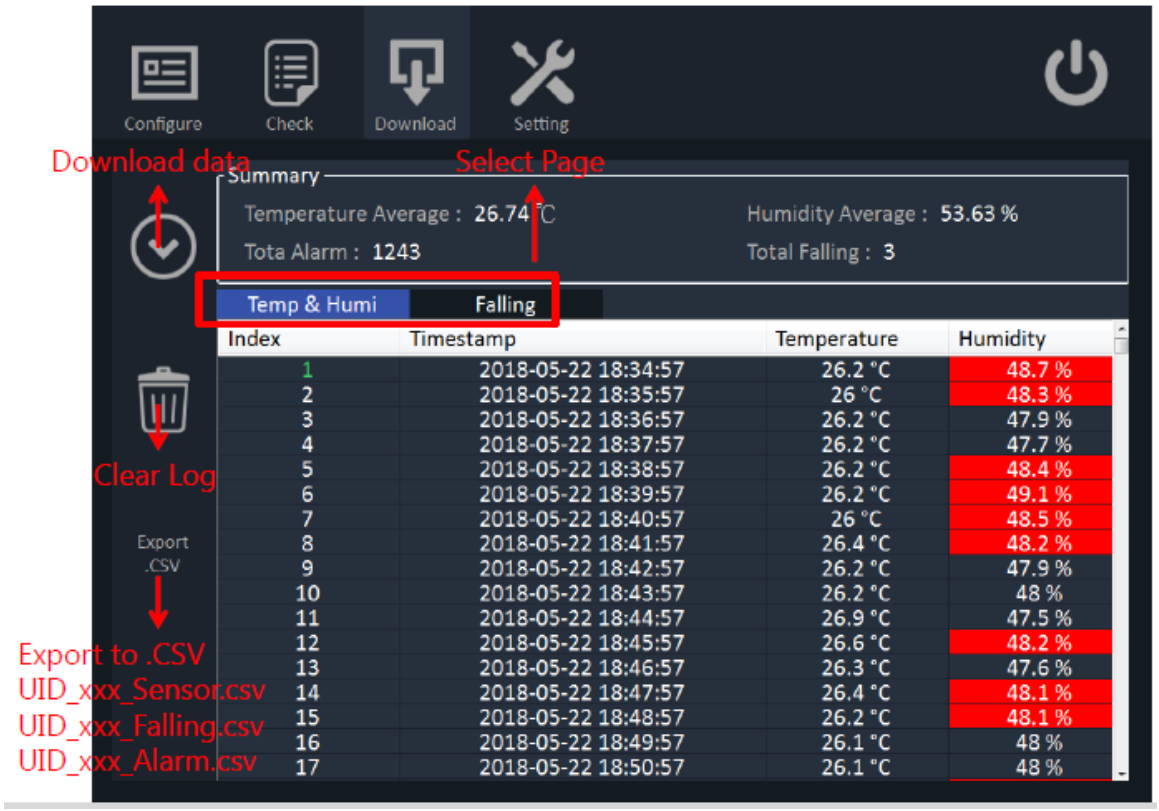
2.2.2 Setup TREK-120 Temperature/humidity sensor

a. Put TREK-120 on the NFC reader. (Note : Please check NFC logo on TREK-120 close to NFC reader 4 LED area to ensure better transmission quality.)

b. Go to " Configure" to setup TREK-120. When you finish all the setting. Please press "Configure Loggers". All the setting will be written into TREK-120. Press " Start logger" , then the TREK-120 will start record temperature/humidity. Press " Stop logger", then the TREK-120 will be stopped record.

c. Go to "Check" to check the configuration of the TREK-120.

d. Go to “Download” to download all data in the TREK-120. When you press “ Download data”, all data will be shown on the screen. You also can export to .CSV file to save in other disk.

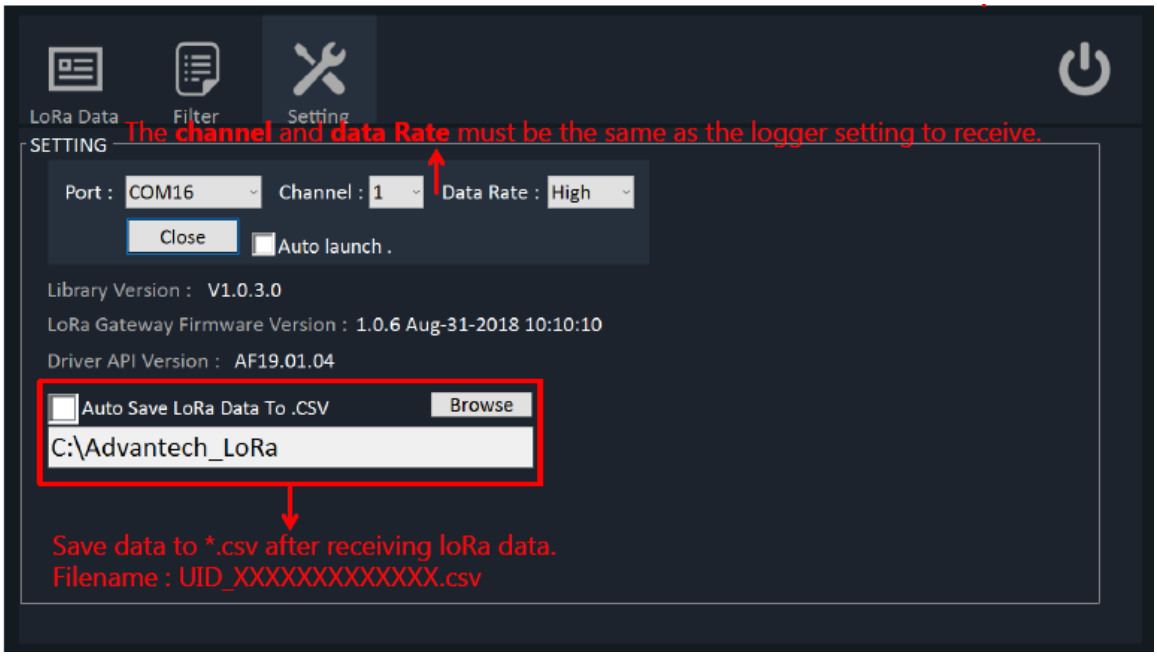


2.2.3 Setup TREK-120 LoRa gateway (Windows Platform)

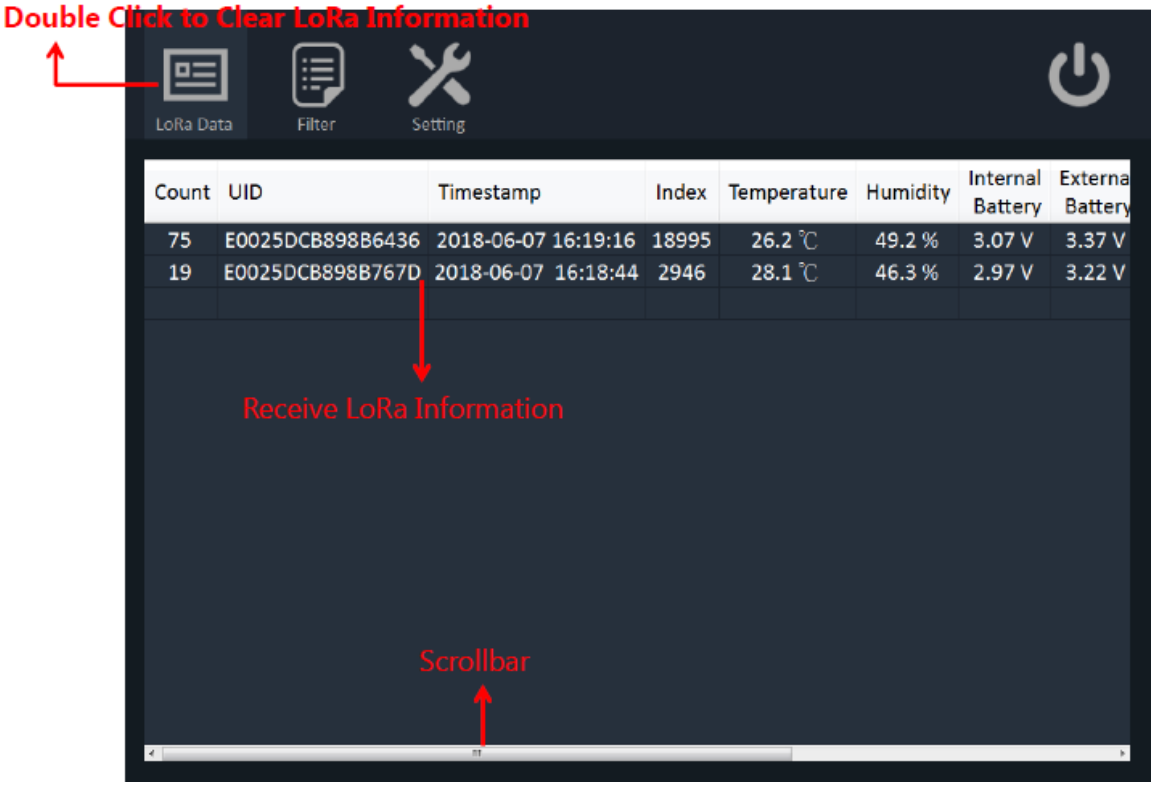
a. Connect TREK-120 LoRa gateway USB cable to the PC or any windows platform USB port.

b. Go to “Setting”.

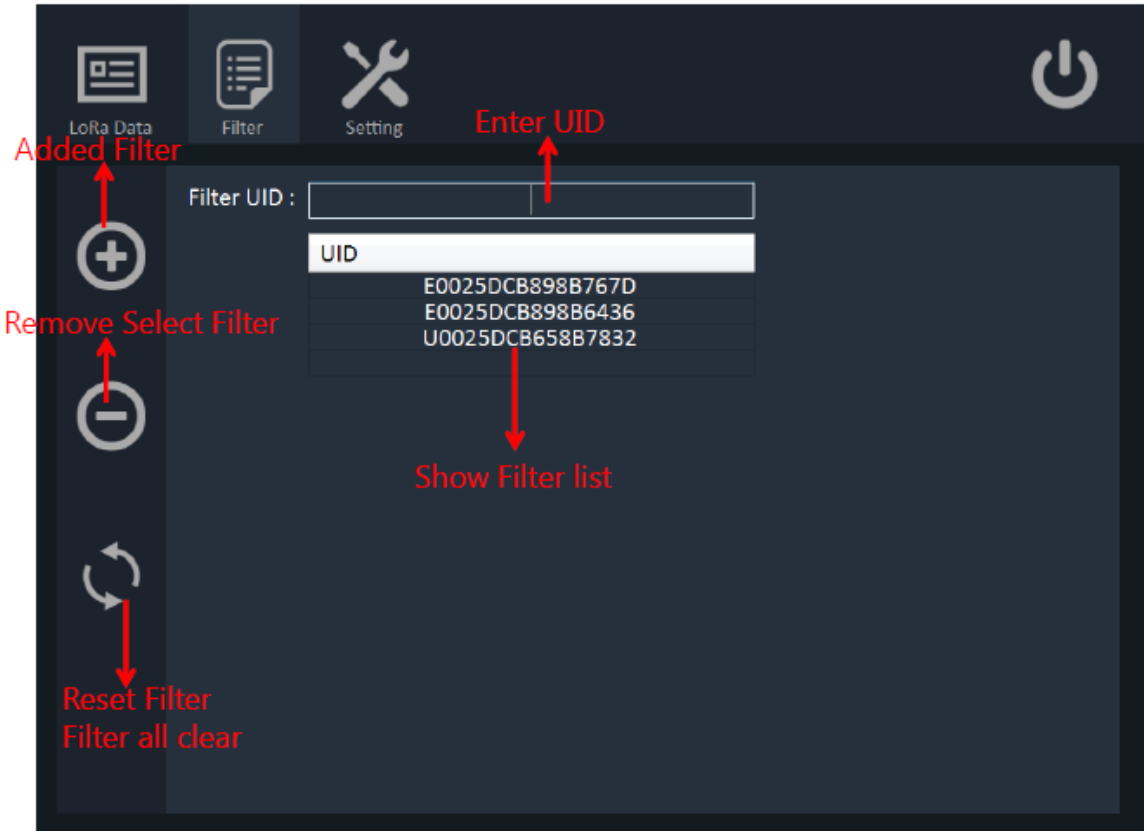
c. Select COM port which LoRa gateway connected and channel & Data rate. (Note : The channel and data rate must be the same as the TREK-120 sensor setting.) Then press “Open”.



d. Go to “LoRa data” then will show all the data.



e. Go to “Filter”. You can select UID to show which TREK-120 you want to see the data.


LoRa Data
Filter
Setting
Enter UID
Added Filter
Filter UID :
UID
E0025DCB898B767D
E0025DCB898B6436
U0025DCB658B7832
Remove Select Filter
Show Filter list
Reset Filter
Filter all clear

[鍵入文字]

Chapter 3

Cold Chain API

This chapter explains the software API for TREK-120

3.1 Introductions

SDK Overview

The SDK is divided into three parts
BIN
DLL
Sample code
DLL provides logic to communicate with the Logger module, most of the developers do not need to know the details, only need to corresponding process after the interest event trigger or Read/Write logger sensor.

System Requirement

OS
☐ Windows 7/8/10

Software
☐ Visual Studio 2015
☐ .NET Framework 4.0+

Language
☐ C# only

SDK Package Structure

You can get the following contents by extracting the SDK package:

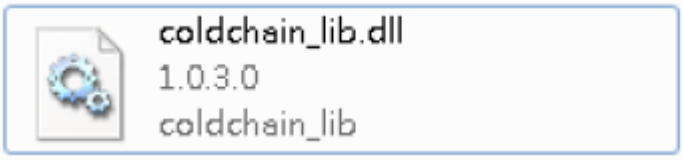
Directory	Description
bin	DLL & Demo Program
doc	Document and Release Note
include	RFID Driver
sample	The SDK sample source codes

Using Cold Chain Library

Start Cold Chain
First Step user need to add a reference in Visual C#
1. In Solution Explorer, right-click the project node and click **Add Reference**.

2. In the Add Reference dialog box, select the tab indicating the type of component you want to reference.

3. Select the **coldchain_lib.dll**, and then click OK.



4. Add Namespaces to your code

```
using coldchain_lib;
```

```
//NFC Reader API Function
ColdChainControl.Reader ColdChain_API = new ColdChainControl.Reader();
```

```
//Gateway API Function
ColdChainControl.Gateway ColdChain_Gateway_API = new ColdChainControl.Gateway();
```

When the instance is created, the user needs to call the coldchain_init(COM_Port) or coldchain_gateway_init(COM_Port) method to open the corresponding Serial Port. When the module is turned on, the system will be initialized in the background. When the initialization is complete, the user should be able to get the firmware version through coldchain_get_version() or coldchain_gateway_get_version(). If the version return "Unknown", try again for a while, or check that your device is properly connected and the power is on.

3.2 Application Programming Interface(API)

Enumeration

AlarmMode Enum

☐ (0) **CONTINUOUS** - Alarm trigger mode to continuous.

☐ (1) **CUMULATIVE** - Alarm trigger mode to cumulative.

ButtonActivate Enum

☐ (0) **NOT_SUPPORT** - The not support button function.

☐ (1) **SUPPORT** - The support button function.

SensorState Enum

☐ (0) **CLOSE_NOT_SETTING** - The Logger sensor not setting and not start.

☐ (1) **CLOSE_ALREADY_SETTING** - The Logger sensor already setting and not start.

☐ (2) **START_RECORD** - The Logger sensor start up.

WarningType Enum

☐ (0) **NORMAL** - The temperature and humidity alarm normal status.

☐ (1) **HIGH** - The temperature and humidity alarm upper status.

☐ (2) **LOW** - The temperature and humidity alarm lower status.

Structure / Classes

Version Structure

Syntax:

Windows:

```
public struct Version
{
        public string lib_version;
        public string device_firmware_version;
        public string driver_api_version;
}
```

Description:

Get the version of firmware.

Members:

lib_version

Point to coldchain_lib Version.

device_firmware_version Point to NFC Reader OR LoRa Gateway Version.

driver_api_version

Point to Jogtek RFID.DLL API Version.

Setting_Logger Structure

Syntax:

Windows:

```
public struct Setting_Logger
{
        public int record_interval_minutes;

        public int temperature_alarm_upper_bound;
        public int temperature_alarm_lower_bound;
        public int temperature_alarm_frequency;
        public AlarmMode temperature_alarm_mode;

        public ButtonActivate button_activate;
        public ButtonActivate button_datafull_restart;

        public int humidity_alarm_upper_bound;
        public int humidity_alarm_lower_bound;
        public int humidity_alarm_frequency;
        public AlarmMode humidity_alarm_mode;

        public int falling_alarm_bound;
        public double external_battery_stop_logger;
        public bool activate_lora_transmission;
        public bool activate_falling_record;
        public StartTime appointment_time;

        public bool loop_recording;
        public int lora_channel;
        public int lora_data_rate;
}
```

Description:

Configure logger sensor internal information.

Members:

record_interval_minutes

Set Logger sensor record interval (minutes).

temperature_alarm_upper_bound

Set logger sensor temperature alarm upper.

temperature_alarm_lower_bound

Set logger sensor temperature alarm lower.

temperature_alarm_frequency

Set logger sensor temperature alarm frequency.

temperature_alarm_mode

Temperature according to alarm frequency select type.
Please refer to AlarmMode Enum.

button_activate

Press button to start logger sensor.
Please refer to ButtonActivate Enum.

button_datafull_restart

Press button to restart logger sensor when record data full.
Please refer to ButtonActivate Enum.

humidity_alarm_upper_bound

Set logger sensor humidity alarm upper.

humidity_alarm_lower_bound

Set logger sensor humidity alarm lower.

humidity_alarm_frequency

Set logger sensor humidity alarm frequency.

humidity_alarm_mode

humidity according to alarm frequency select type.
Please refer to AlarmMode Enum.

falling_alarm_bound

Logger sensor falling alarm bound.

external_battery_stop_logger

The external battery voltage is lower than the set value to stop LoRa.

activate_lora_transmission

Enable/Disable lora transmission function.

activate_falling_record

Enable/Disable falling record function.

appointment_time

Appointment time to start logger, Time format "Date / Hour / Minute / Second".
Please refer to StartTime Structure.

loop_recording

LoRa Extend to 60K message and overwrite.

lora_channel

LoRa frequency channel, Value 1 to 5.

lora_data_rate

Data Rate Value : 0 is Low , 1 is High.

Sensor_Internal_Information Structure

Syntax:

Windows:

```
public struct Sensor_Internal_Information
{
    public string uid;
    public ButtonActivate button_activate;
    public DateTime first_record_datetime;
    public int record_index;
    public int record_interval_minutes;

    public int temperature_alarm_upper_bound;
    public int temperature_alarm_lower_bound;
    public int temperature_alarm_frequency;
    public AlarmMode temperature_alarm_mode;

    public SensorState sensor_status;
    public double internal_battery_voltage;
    public double external_battery_voltage;
    public string sensor_firmware_version;

    public int humidity_alarm_upper_bound;
    public int humidity_alarm_lower_bound;
    public int humidity_alarm_frequency;
    public AlarmMode humidity_alarm_mode;

    public int falling_record_index;
    public int falling_alarm_bound;
    public Overflow_Flag over_alert;

    public int lora_channel;
    public int lora_data_rate;

      public bool loop_recording;
      public double external_battery_stop_logger;
      public string external_battery_life;

      public bool activate_lora_transmission;
      public bool activate_falling_record;
      public StartTime appointment_time;
      public int transmission_failed;
}
```

Description:

Get logger sensor internal information.

Members:

uid

The logger sensor UID number.

button_activate Button start or restart logger sensor.

Please refer to ButtonActivate Enum.

first_record_datetime

Logger sensor the first record time stamp.

record_index

Logger sensor record count.

record_interval_minutes

Logger sensor record interval (minutes).

temperature_alarm_upper_bound

Logger sensor temperature alarm upper.

temperature_alarm_lower_bound

Logger sensor temperature alarm lower.

temperature_alarm_frequency

Logger sensor temperature alarm frequency.

temperature_alarm_mode

Temperature according to alarm frequency select type.

Please refer to AlarmMode Enum.

sensor_status

logger sensor status.

Please refer to SensorState Enum.

internal_battery_voltage

Internal battery voltage.

external_battery_voltage

External battery voltage.

sensor_firmware_version

Logger sensor firmware version.

humidity_alarm_upper_bound

Logger sensor humidity alarm upper.

humidity_alarm_lower_bound

Logger sensor humidity alarm lower.

humidity_alarm_frequency

Logger sensor humidity alarm frequency.

humidity_alarm_mode

Humidity according to alarm frequency select type.

Please refer to AlarmMode Enum.

falling_record_index

Logger sensor falling count.

falling_alarm_bound

Logger sensor falling alarm bound.

over_alert

The LoRA sensor overflow flag.

Please refer to Overflow_Flag Structure.

lora_channel

Get LoRa frequency channel.

lora_data_rate

Get Data Rate Value : 0 is Low , 1 is High.

loop_recording

LoRa Extend to 60K message and overwrite

external_battery_stop_logger

The external battery voltage is lower than the set value to stop LoRa.

external_battery_life

External Battery life (10% ~ 100%).

activate_lora_transmission

Get LoRa Transmission function status.

activate_falling_record

Get Falling Rrecord function status.

appointment_time

Get Appointment Time status.

Please refer to StartTime Structure.

transmission_failed

Logger did not transmit due to collision.

Sensor_Internal_data Structure

Syntax:

Windows:

```
public struct Sensor_Internal_data
{
        public string uid;
        public int record_index;
        public DateTime[] record_datetime;

        public double[] temperature;
        public int temperature_alarm_upper_bound;
        public int temperature_alarm_lower_bound;
        public WarningType[] temperature_Warning;

        public double[] humidity;
        public int humidity_alarm_upper_bound;
        public int humidity_alarm_lower_bound;
        public WarningType[] humidity_Warning;

        public double internal_battery_voltage;
        public double external_battery_voltage;

        public int falling_record_index;
        public DateTime[] falling_datetime;
        public int total_alarm_count;
}
```

Description:

Get logger sensor record temperature and humidity information.

Members:

uid

The logger sensor UID number.

record_index

Logger sensor record count.

record_datetime[]

One array.

Record each data timestamp.

temperature[]

One array.

Record each data temperature.

temperature_alarm_upper_bound

logger sensor temperature alarm upper.

temperature_alarm_lower_bound

logger sensor temperature alarm lower.

temperature_Warning

Judgment temperature alarm status.

Please refer to WarningType Enum.

humidity[]

One array.

Record each data humidity.

humidity_alarm_upper_bound

logger sensor humidity alarm upper.

humidity_alarm_lower_bound

logger sensor humidity alarm lower.

humidity_Warning

Judgment humidity alarm status.

Please refer to WarningType Enum.

internal_battery_voltage

Internal battery voltage.

external_battery_voltage

External battery voltage.

falling_record_index

Logger sensor falling count.

falling_datetime[]

One array.

Record each falling data timestamp.

total_alarm_count

Temperature and Humidity total alarm count.

Sensor_LoRa_data Structure

Windows:

```
public struct Sensor_LoRa_data
{
        public string uid;
        public int record_index;
        public string record_datetime;

        public double temperature;
        public double humidity;

        public double internal_battery_voltage;
        public double external_battery_voltage;
        public int falling_record_index;
        public string falling_datetime;

        public int rssi;
        public int snr;
        public int crc;
        public string external_battery_life;
        public string firmware_version;
        public Overflow_Flag over_alert;
}
```

Description:

Receive LoRa gateway information Structure.

Members:

uid

The loRa sensor UID number.

record_index

LoRa sensor record count.

record_datetime

Receive data timestamp.

temperature

Receive data temperature.

humidity

Receive data humidity.

internal_battery_voltage

Internal battery voltage.

external_battery_voltage

External battery voltage.

falling_record_index

LoRa sensor falling count.

falling_datetime

Receive falling data timestamp.

rssi

The received signal strength indication.

snr

The signal to noise ratio.

crc

The cyclic redundancy check.

external_battery_life

Shows the percentage of external voltage.

firmware_version

LoRa sensor version.

over_alert

The LoRA sensor overflow flag.

Please refer to Overflow_Flag Structure.

Overflow_Flag Structure

Windows:

```
public struct Overflow_Flag
{
        public bool temperature_overflow;
        public bool humidity_overflow;
        public bool falling_overflow;
        public bool record_index_overflow;
        public bool falling_record_index_overflow;
        public bool lora_transmission_stop;
        public bool internal_battery_low;
        public bool external_battery_low;;
}
```

Description:

Save the LoRA sensor overflow flag.

Members:

temperature_overflow

The temperature has exceeded the limit.

humidity_overflow

The humidity is above the limit.

falling_overflow

The number of falling exceeds (or equals) the upper limit.

record_index_overflow

The number of temperature and humidity pens has reached 2000.

falling_record_index_overflow

The number of falling pens has reached 30.

lora_transmission_stop

The external battery voltage is lower than the voltage configuration value before wireless transmission.

internal_battery_low

The internal battery voltage has been lower than 2.4v in front of the wireless transmission

external_battery_low

Set the external voltage limit, stop LoRa transmission.

StartTime Structure

Windows:

```
public struct StartTime
{
        public bool auto_startup;
        public int date;
        public int hour;
        public int minute;
        public int second;
}
```

Description:

Save the LoRA sensor overflow flag.

Members:

auto_startup

Enable / Disable Appointment Time Function.

date

Setting Date time.

hour

Setting Hour time.

minute

Setting Minute time.

second

Setting Second time.

APIs

NFC Reader

coldchain_init

Syntax:

Windows	public int **coldchain_init**(string COM_Port)

Description:

Open the NFC Reader through the serial port.

Parameters:

COM_Port [in]

Specifies which Serial Port Path to connect firmware. For example "COM1"

Returns:

ADVCC_ERR_SUCCESS - On success.

Otherwise see the error code list.

coldchain_deinit

Syntax:

Windows	public int **coldchain_deinit**()

Description:

Close NFC Reader the serial port.

Parameters:

None

Returns:

ADVCC_ERR_SUCCESS - On success.

Otherwise see the error code list.

coldchain_get_version

Syntax:

Windows	public int **coldchain_get_version**(ref Version FirmwareVer)

Description:

Get the version of firmware.

Parameters:

FirmwareVer [out]

Please refer to Version Structure.

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

None

coldchain_set_logger

Syntax:

Windows	public int **coldchain_set_logger**(Setting_Logger Logger)

Description:

Configure logger sensor internal information.

Parameters:

Logger [in]

Please refer to Setting_Logger Structure.

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

The **lora_channel** and **lora_data_rate** must be the same as the Gateway setting to receive.

coldchain_start_logger

Syntax:

Windows	public int **coldchain_start_logger**()

Description:

Start logger sensor to record data.

Parameters:

None

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

None

coldchain_stop_logger

Syntax:

Windows	public int **coldchain_stop_logger**()

Description:

Stop logger sensor.

Parameters:

None

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

None

coldchain_get_sensor_info

Syntax:

Windows	public int **coldchain_get_sensor_info**(ref Sensor_Info information)

Description:

Get logger sensor internal information.

Parameters:

information [out]

Please refer to Sensor_Internal_Information Structure.

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

None

coldchain_get_sensor_data

Syntax:

Windows	public int **coldchain_get_sensor_data**(ref Sensor_data OD_data)

Description:

Get logger sensor record temperature and humidity information.

Parameters:

OD_data [out]

Please refer to Sensor_Internal_data Structure.

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

None

coldchain_get_temperature_avg

Syntax:

Windows	public int **coldchain_get_temperature_avg**(Sensor_data OD_data, ref double temperature_avg)

Description:

Get temperature average.

Parameters:

OD_data [in]

Pointer to the node object handle that is the returned by coldchain_get_sensor_data().

temperature_avg [out]

Calculation total temperature average.

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

None

coldchain_get_humidity_avg

Syntax:

Windows	public int **coldchain_get_humidity_avg**(Sensor_data OD_data, ref double humidity_avg)

Description:

Get humidity average.

Parameters:

OD_data [in]

Pointer to the node object handle that is the returned by coldchain_get_sensor_data().

humidity_avg [out]

Calculation total humidity average.

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

None

coldchain_export_humiture_data

Syntax:

Windows	public int **coldchain_export_humiture_data**(Sensor_data OD_data, string FilePath)

Description:

Export temperature and humidity data to csv.

Parameters:

OD_data [in]

Pointer to the node object handle that is the returned by coldchain_get_sensor_data().

FilePath [in]

Save file location.

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

Save file to .csv format.

coldchain_export_falling_data

Syntax:

Windows	public int **coldchain_export_falling_data**(Sensor_data OD_data, string FilePath)

Description:

Export falling data to csv.

Parameters:

OD_data [in]

Pointer to the node object handle that is the returned by coldchain_get_sensor_data().

FilePath [in]

Save file location.

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

Save file to .csv format.

coldchain_export_alarm_data

Syntax:

Windows	public int **coldchain_export_alarm_data**(Sensor_data OD_data, string FilePath)

Description:

Export temperature and humidity alarm data to csv.

Parameters:

OD_data [in]

Pointer to the node object handle that is the returned by coldchain_get_sensor_data().

FilePath [in]

Save file location.

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

Save file to .csv format.

LoRa Gateway

coldchain_gateway_init

Syntax:

Windows	public int **coldchain_gateway_init**(string COM_Port, int lora_channel, int lora_data_rate)

Description:

Open the LoRa Gateway through the serial port.

Parameters:

COM_Port [in]

Specifies which Serial Port Path to connect firmware. For example "COM1"

lora_channel [in]

Set LoRa frequency channel, Value 1 to 5.

lora_data_rate [in]

Set Data Rate. Value : 0 is Low , 1 is High.

Returns:

ADVCC_ERR_SUCCESS - On success.

Otherwise see the error code list.

Remarks:

The **lora_channel** and **lora_data_rate** must be the same as the logger setting to receive.

coldchain_gateway_deinit

Syntax:

Windows	public int **coldchain_gateway_deinit**()

Description:

Close LoRa Gateway the serial port.

Parameters:

None

Returns:

ADVCC_ERR_SUCCESS - On success.

Otherwise see the error code list.

coldchain_gateway_get_version

Syntax:

Windows	public int **coldchain_gateway_get_version**(ref Version FirmwareVer)

Description:

Get the Gateway version of firmware.

Parameters:

FirmwareVer [out]

Please refer to Version Structure.

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

None

coldchain_gateway_start_receive_lora_data

Syntax:

CallBack Function	public delegate void **ReceiveLoRaCallBack**(Sensor_LoRa_data data);
Windows	public int **coldchain_gateway_start_receive_lora_data**(ReceiveLoRaCallBack cb)

Description:

When the event is triggered, the ReceiveLoRaCallBack() callback of registered handler instance will be triggered.

Parameters:

cb [in]

Pointer to a user defined callback function.

data[out]

Please refer to Sensor_LoRa_data Structure.

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

None

coldchain_gateway_stop_receive_lora_data

Syntax:

Windows	public int **coldchain_gateway_stop_receive_lora_data**()

Description:

Clear the callback registration of the ReceiveLoRaCallBack.

Parameters:

None

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

None

coldchain_gateway_added_filter

Syntax:

Windows	public int **coldchain_gateway_added_filter**(string uid)

Description:

Designate LoRa uid add whitelist for filter, Only added uid will receive.

Parameters:

uid[in]

Receive LoRa uid.

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

None

coldchain_gateway_remove_filter

Syntax:

Windows	public int **coldchain_gateway_remove_filter**(string uid)

Description:

Designate LoRa uid remove whitelist.

Parameters:

uid[in]

Receive LoRa uid.

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

None

coldchain_gateway_reset_filter

Syntax:

Windows	public int **coldchain_gateway_reset_filter**()

Description:

Reset whitelist.

Parameters:

None

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

None

coldchain_gateway_get_filter

Syntax:

Windows	public int **coldchain_gateway_get_filter**(ref string[] uid)

Description:

Get all the uid values in the whitelist.

Parameters:

uid [out]

Get setting uid list..

Returns:

MRM_ERR_NO_ERROR - On success.

Otherwise see the error code list.

Remark:

None

Error Code List

- (0) **ADVCC_ERR_SUCCESS** - On success.
- (1) **ADVCC_ERR_GET_VERSION_FAILED**
- (2) **ADVCC_ERR_OPERATION_FAILED**
- (3) **ADVCC_ERR_EXPORT_DATA_FAILED**
- (7) **ADVCC_ERR_COM_IS_OPEN**
- (8) **ADVCC_ERR_COM_NOT_OPEN**
- (9) **ADVCC_ERR_COM_PORT_ERROR**
-
- (20000000) **ADVCC_ERR_SET_SENSOR_FAILED**
- (20000001) **ADVCC_ERR_SET_INTERVAL_ERROR**
- (20000002) **ADVCC_ERR_SET_TEMP_UPPER_ERROR**
- (20000003) **ADVCC_ERR_SET_TEMP_LOWER_ERROR**
- (20000004) **ADVCC_ERR_SET_TEMP_FREQUENCY_ERROR**
- (20000005) **ADVCC_ERR_SET_HUMIDITY_UPPER_ERROR**
- (20000006) **ADVCC_ERR_SET_HUMIDITY_LOWER_ERROR**
- (20000007) **ADVCC_ERR_SET_HUMIDITY_FREQUENCY_ERROR**
- (20000008) **ADVCC_ERR_SET_FALLING_BOUND_ERROR**
- (20000009) **ADVCC_ERR_SET_EXTERNAL_BATTERY_STOP_LOGGER_ERROR**
- (20000010) **ADVCC_ERR_SET_APPOINTMENT_TIME_ERROR**
- (20000011) **ADVCC_ERR_SET_LORA_CHANNEL_ERROR**
-
- (30000000) **ADVCC_ERR_GET_SENSOR_FAILED**
- (30000001) **ADVCC_ERR_GET_DATA_FAILED**
- (30000002) **ADVCC_ERR_GET_FALLING_FAILED**
- (30000003) **ADVCC_ERR_GET_AVG_FAILED**
-
- (40000000) **ADVCC_ERR_GATEWAY_CALLBACK_FAILED**
- (40000001) **ADVCC_ERR_GATEWAY_RECEIVE_FAILED**
- (40000002) **ADVCC_ERR_GATEWAY_RECEIVE_ACTIVE**

- (40000003) **ADVCC_ERR_GATEWAY_RECEIVE_NOT_ACTIVE**
- (40000004) **ADVCC_ERR_GATEWAY_FILTER_ALREADY_SETTING**
- (40000005) **ADVCC_ERR_GATEWAY_FILTER_NOT_SETTING**

Trusted ePlatform Services

ADVANTECH

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.
No part of this publication may be reproduced in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission of the publisher.
All brand and product names are trademarks or registered trademarks of their respective companies.
© Advantech Co., Ltd. 2010