# IoT Controller

## Featuring LoRaWAN®

## UC100

User Guide

## Safety Precautions

Milesight will not shoulder responsibility for any loss or damage resulting from not following the instructions of this operating guide.

- The device must not be remodeled in any way.
- Do not place the device close to objects with naked flames.
- Do not place the device where the temperature is below/above the operating range.
- Power off the device when installing or wiring.
- Make sure electronic components do not drop out of the enclosure while opening.
- The device must never be subjected to shocks or impacts.

## Declaration of Conformity

UC100 is in conformity with the essential requirements and other relevant provisions of the CE, FCC, and RoHS.




For assistance, please contact
Milesight technical support:
Email: iot.support@milesight.com
Support Portal: support.milesight-iot.com
Tel: 86-592-5085280
Fax: 86-592-5023065
Address: Building C09, Software Park III, Xiamen 361024, China

## Revision History

| Date | Doc Version | Description |
|---|---|---|
| May 27, 2022 | V 1.0 | Initial version |
| Dec. 5, 2022 | V 1.1 | Add active pass-through feature and two-way pass-through feature |
| Jan. 24, 2024 | V 1.2 | 1. Add data storage, retransmission and retrievability feature<br>2. Increase to 32 Modbus channels<br>3. Add downlink commands to configure Modbus channels<br>4. Add Modbus channel alarm feature |
| Jan. 7, 2025 | V 1.3 | 1. Add optional power converter kit accessory<br>2. Support to report sign type of Modbus channels<br>3. Add Modbus channel change alarm report<br>4. Add device class type, TSL version and reset report |

# Contents

# 1. Product Introduction

## 1.1 Overview

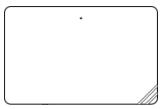
UC100 is an IoT controller used for remote control and data acquisition from Modbus RS485 devices via LoRaWAN® networks. It can read up to 32 Modbus RTU devices and support Modbus transparent transmission between server and RS485 devices as a Modbus to LoRaWAN® converter. Besides, UC100 supports multiple trigger conditions and actions which can work autonomously even when the network drops.

## 1.2 Features

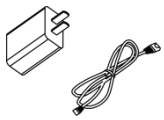
- Easy to connect with diverse wired sensors through RS485 interfaces
- Support LoRaWAN® wireless communication
- Multiple triggering conditions and actions
- Embedded watchdog for work stability
- Industrial metal case design with a wide operating temperature range
- Compliant with standard LoRaWAN® gateways and network servers
- Quick and easy management with Milesight IoT Cloud or Milesight Development Platrorm solution

# 2. Hardware Introduction

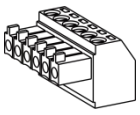
## 2.1 Packing List



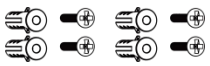
1 × UC100 Device



1 × Type-C Cable (1.2m) & Power Adapter

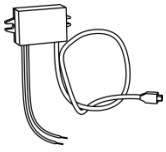


1 × Terminal Block



4 × Wall Mounting Kits

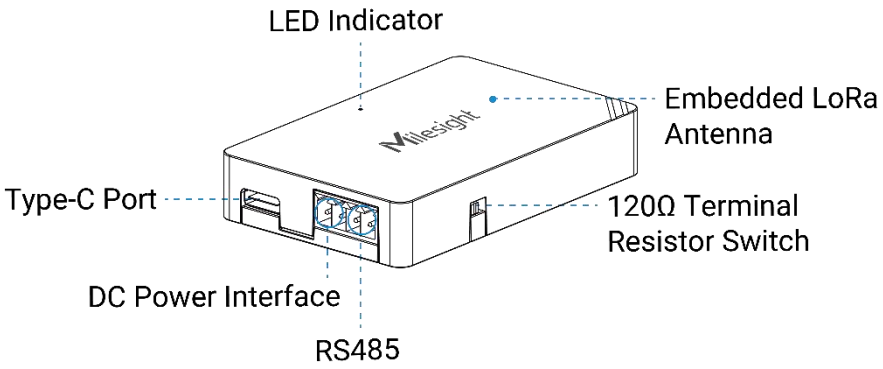

1 × Quick Guide



1 × AC/DC-DC Power Converter Kit (Optional)

**If any of the above items are missing or damaged, please contact your sales representative.**

## 2.2 Hardware Overview



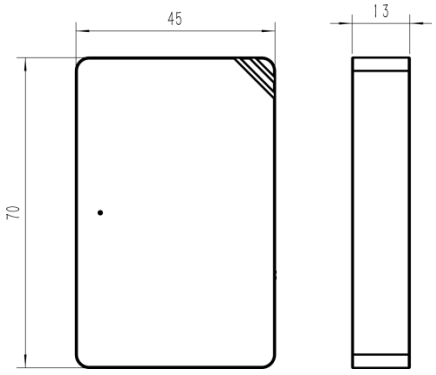

**120Ω Terminal Resistor Switch**: the device will add a 120Ω termination resistor to avoid data-corrupting reflections if RS485 data rate is high or cable length is long.

## 2.3 LED and Reset Button

The reset button is inside the device.

| Function | Action | LED Indication |
|---|---|---|
| Work Status | System is functioning properly | Static On |
| | Fail to acquire data from data interfaces | Slowly Blinks |
| | Device upgrade or system error | Static On |
| Reboot | Press and hold the button for more than 3 seconds. | Slowly Blinks |
| Reset | Press and hold the button for more than 10 seconds. | Quickly Blinks |

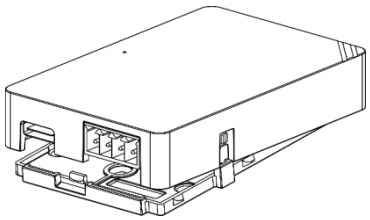
## 2.4 Dimensions (mm)



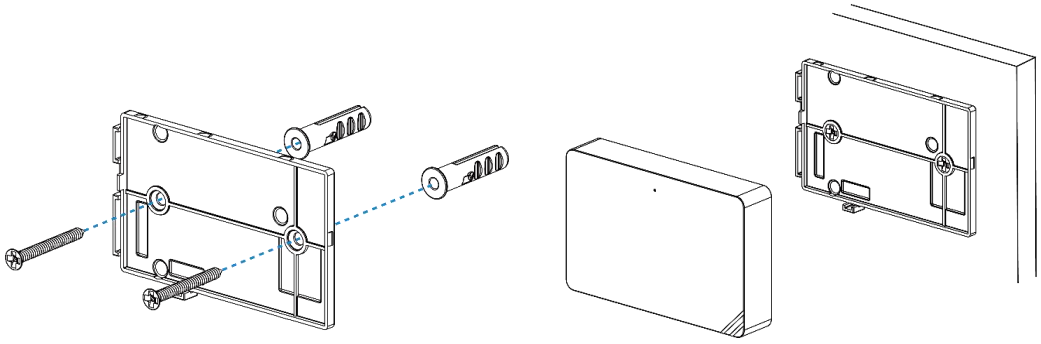

## 3. Device Installation

UC100 device can be placed on a desktop or mounted to a wall.

1. Take off the back cover of UC100 device, and fix the wall plugs into the wall according to the drilling position as referred.



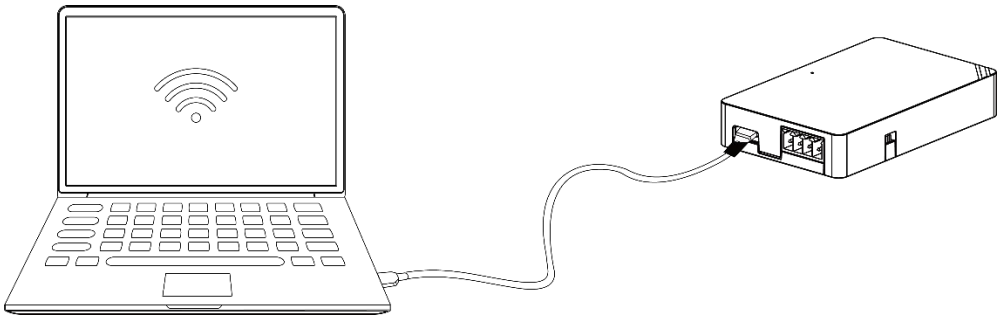
2. Screw the cover on the mounting positions and install back the device.



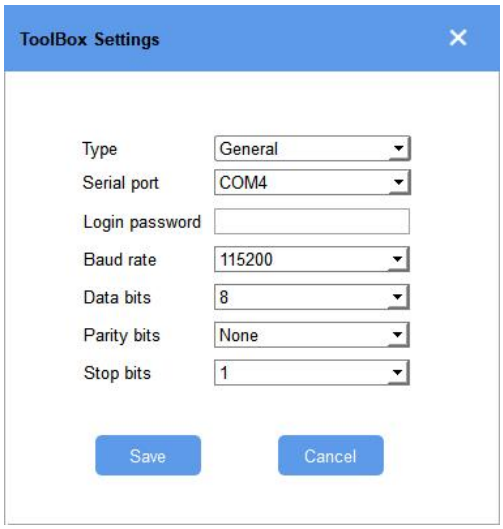
# 4. Operation Guide

## 4.1 Log in the ToolBox

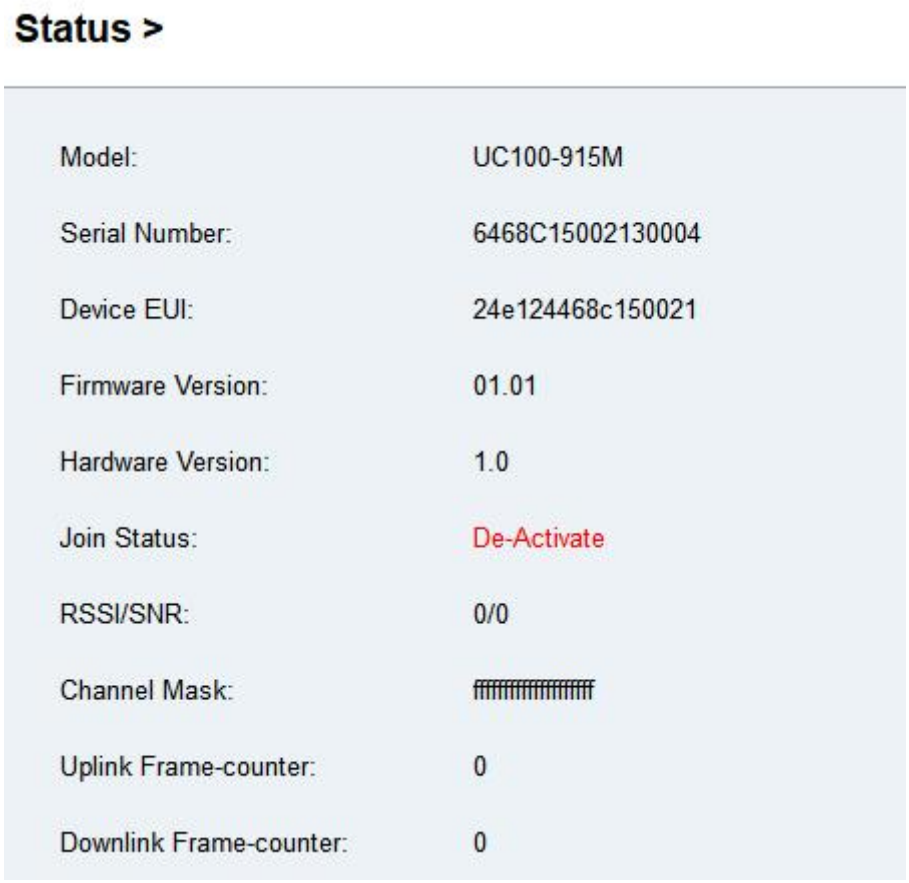
1. Download ToolBox software from Milesight website.
2. Power on the UC100 device, then connect it to computer via the type-C port.



3. Open the ToolBox and select type as **General**, then click password to log in ToolBox. (Default password: **123456**)




4. After logging in the ToolBox, you can change device settings.

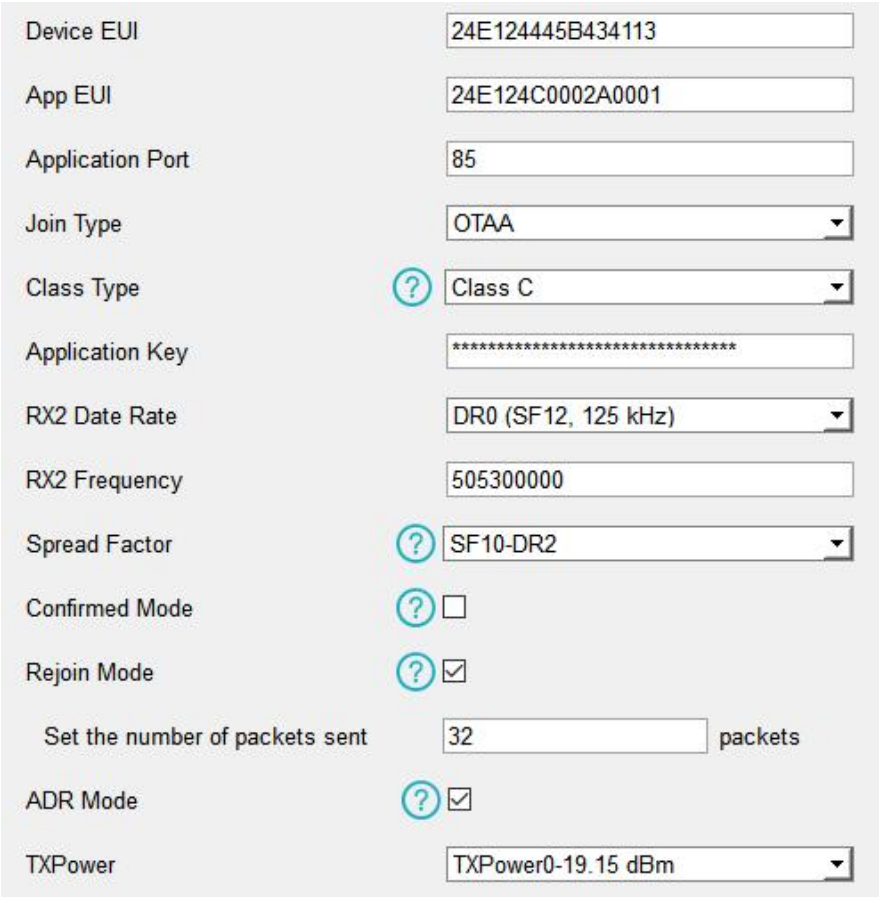



## 4.2 LoRaWAN Settings

LoRaWAN settings are used for configuring the transmission parameters in LoRaWAN® network.

**Basic LoRaWAN Settings:**

Configure join type, App EUI, App Key and other information. You can also keep all settings by default.




| Parameters | Description |
|---|---|
| Device EUI | Unique ID of the device on the label. |
| App EUI | Default App EUI is 24E124C0002A0001. |
| Application Port | The port is used for sending and receiving data, the default port is 85. |
| Working Mode | Fixed as Class C. |
| Join Type | OTAA and ABP modes are available. |
| Application Key | Appkey for OTAA mode, default is 5572404C696E6B4C6F52613230313823. |
| Device Address | DevAddr for ABP mode, default is the 5$^{th}$ to 12$^{th}$ digits of SN. |
| Network Session Key | Nwkskey for ABP mode, default is 5572404C696E6B4C6F52613230313823. |
| Application Session Key | Appskey for ABP mode, default is 5572404C696E6B4C6F52613230313823. |
| RX2 Data Rate | RX2 data rate to receive downlinks or send/receive D2D commands. |
| RX2 Frequency | RX2 frequency to receive downlinks or send/receive D2D commands. Unit: Hz |
| Spread Factor | If ADR is disabled, the device will send data via this spread factor. |
| Confirmed Mode | If the device does not receive ACK packet from network server, it will resend data once. |

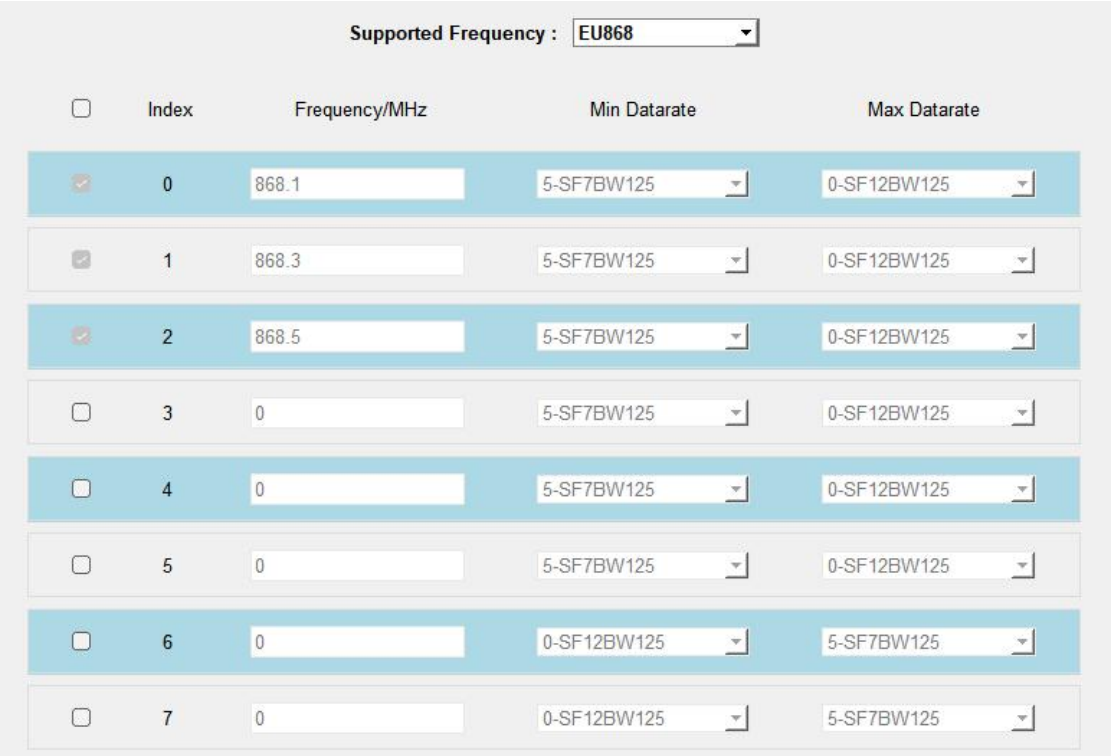
| | |
|---|---|
| Rejoin Mode | Reporting interval ≤ 35 mins: the device will send a specific number of LinkCheckReq MAC packets to the network server every reporting interval or every double reporting interval to validate connectivity; If there is no response, the device will re-join the network.<br>Reporting interval > 35 mins: the device will send a specific number of LinkCheckReq MAC packets to the network server every reporting interval to validate connectivity; If there is no response, the device will re-join the network. |
| Set the number of packets sent | When rejoin mode is enabled, set the number of LinkCheckReq packets sent. |
| ADR Mode | Allow the network server to adjust datarate of the device. |
| Tx Power | The transmit power of device. |

**Note:**

1) Please contact sales for device EUI list if there are many units.
2) Please contact sales if you need random App keys before purchasing.
3) Select OTAA mode if you use Milesight IoT Cloud or Milesight Development Platform to manage devices.
4) Only OTAA mode supports rejoin mode.

### LoRaWAN Frequency Settings:

Go to **LoRaWAN Settings > Channel** to select supported frequency and select channels to send uplinks. Make sure the channels match what you set in the LoRaWAN® gateway.




If the frequency is one of CN470/AU915/US915, enter the index of the channel that you want to enable in the input box, making them separated by commas.

**Examples:**

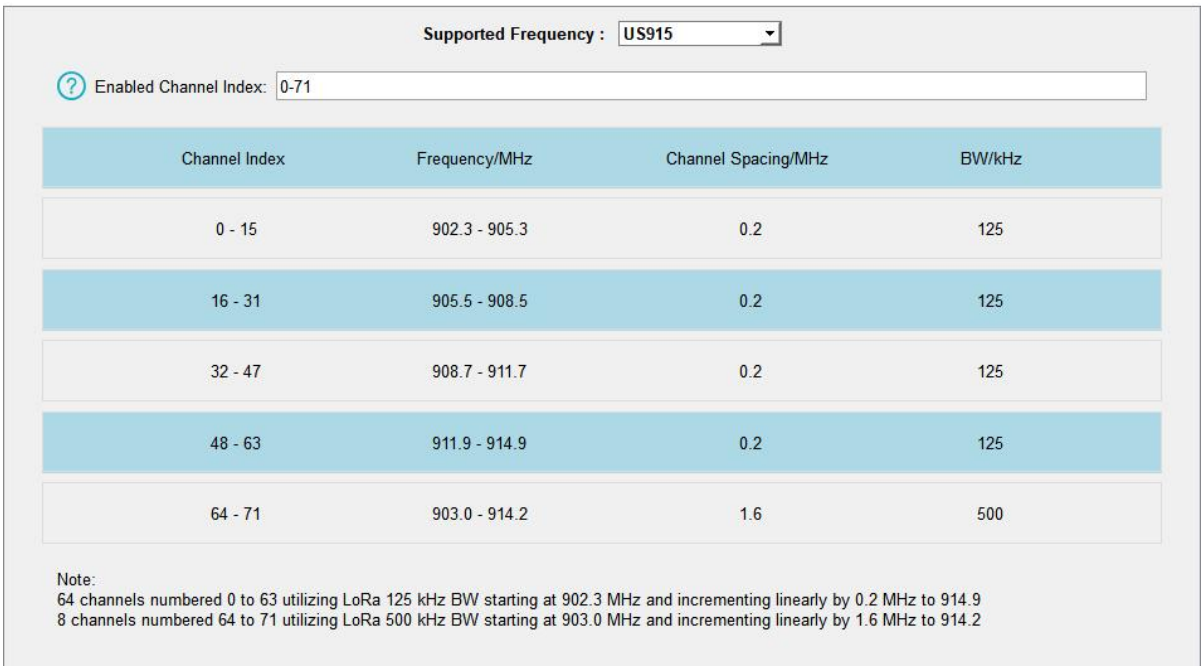
1, 40: Enabling Channel 1 and Channel 40

1-40: Enabling Channel 1 to Channel 40

1-40, 60: Enabling Channel 1 to Channel 40 and Channel 60

All: Enabling all channels

Null: Indicates that all channels are disabled



## 4.3 General Settings

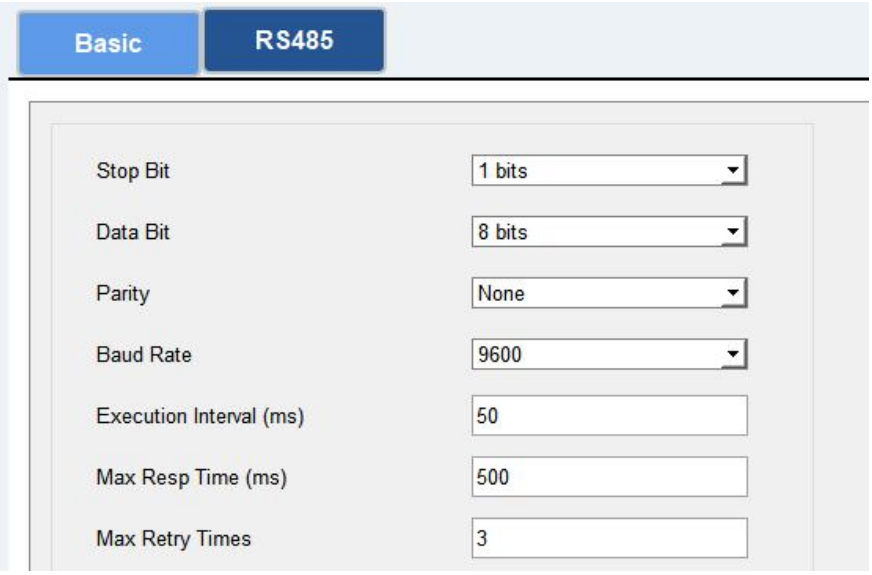

| Parameters | Description |
|---|---|
| Device ID | Show the SN of the device. |
| Reporting Interval | Reporting interval of transmitting Modbus channel data to the network server. Range: 1-1080 mins, default: 20 mins |
| Data Storage | Disable or enable reporting data storage locally. |
| Data Retransmission | Disable or enable data retransmission. |
| D2D | Enable or disable Milesight D2D Feature. |
| D2D Key | Set a unique key the same as the setting in Milesight D2D controller or agent device. The default is 5572404C696E 6B4C6F52613230313823. |
| Change Password | Change the password to log in ToolBox. |

## 4.4 RS485 Settings

UC100 supports to set up communications with RS485 via two ways: Modbus channels or Modbus RS485 bridge LoRaWAN®.

### Basic Serial Settings:

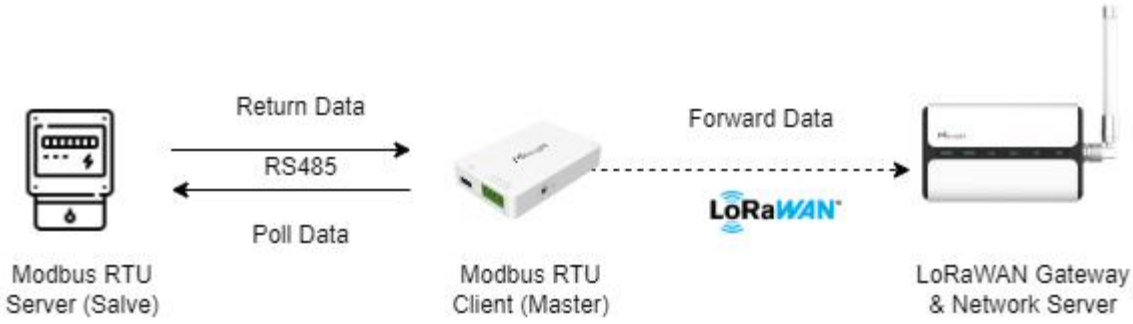
UC100 has one RS485 port for Modbus RTU device connection. The basic serial settings should be the same as RS485 terminal devices.



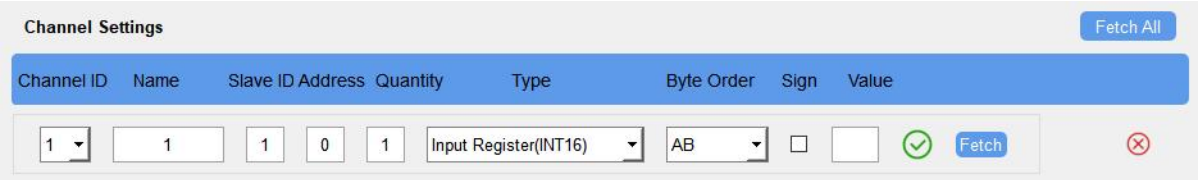

| Parameters | Description |
|---|---|
| Stop Bit | 1 bit/2 bit is available. |
| Data Bit | 8 bit is available. |
| Parity | None, Odd and Even are available. |
| Baud Rate | 1200/2400/4800/9600/19200/38400/57600/115200 are available. |
| Execution Interval (ms) | The execution interval between each Modbus channel command. |
| Max Resp Time (ms) | The maximum response time that the UC100 waits for the reply to the command. If it does not get a response after the max response time, it is determined that the command has timed out. |
| Max Retry Times | Set the maximum retry times after the device fails to read data from RS485 terminal devices. |

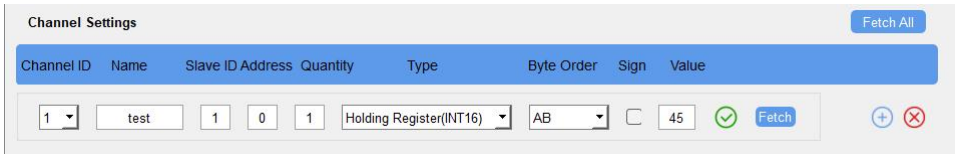
**Modbus Channels:**

UC100 supports to work as a Modbus RTU Client (Master) to poll the data from the RS485 device and return the data to the network server.



Click ⊕ to add Modbus channels, then save configurations.




| Parameters | Description |
|---|---|
| Channel ID | Select the channel ID you want to configure from 32 channels. |
| Name | Customize the name to identify every Modbus channel. |
| Slave ID | Set a Modbus slave ID of a terminal device. |
| Address | The starting address for reading. |
| Quantity | Set read how many digits from the starting address, it fixes to 1. |
| Type | Select the data type of Modbus channels. |
| Byte Order | Set the Modbus data reading order if you configure the type as Input register or holding register.<br>INT32/Float: ABCD, CDBA, BADC, DCBA<br>INT16: AB, BA |
| Sign | The tick indicates that the value has a plus or minus sign. |
| Fetch | Click to send a Modbus read command to test if the RS485 device can reply with the correct values. Note that do not click frequently to avoid the fetch failure due to the slow response of RS485 devices.<br>**Example:** as this setting, the device will send command: 01 03 00 00 00 01 84 0A<br>  |



### Modbus RS485 bridge LoRaWAN®:

UC100 supports to work as a relay to set up the communication between the server and RS485 devices. There are two pass-through modes:

**Active Pass-through:** the network server can send any command to the RS485 device and the RS485 device can only react according to server commands.


**Two-way Pass-through:** not only can network server send any command to the RS485 device,

but also the RS485 device supports transmitting the data to the network server actively.

**Note:** When **Two-way Pass-through** is enabled, Modbus channels can't be used and the corresponding IF-THEN command will not work.



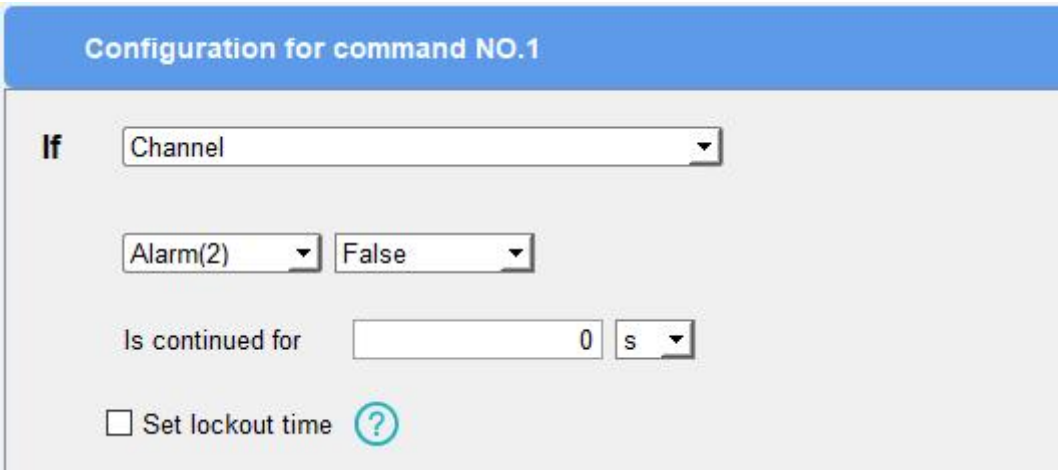
| Parameters | Description |
|---|---|
| Modbus RS485 bridge LoRaWAN® | Enable or disable the Modbus RS485 bridge LoRaWAN® feature. |
| Pass-through Mode | Select from Active Pass-through or Two-way Pass-through. |
| Port | The communication port between the RS485 device and the network server. Range: 2-84, 86-223. |

## 4.5 IF-THEN Command

UC100 supports configuring locally IF-THEN commands to do some actions automatically even without a network connection. One device can be added 16 commands at most.

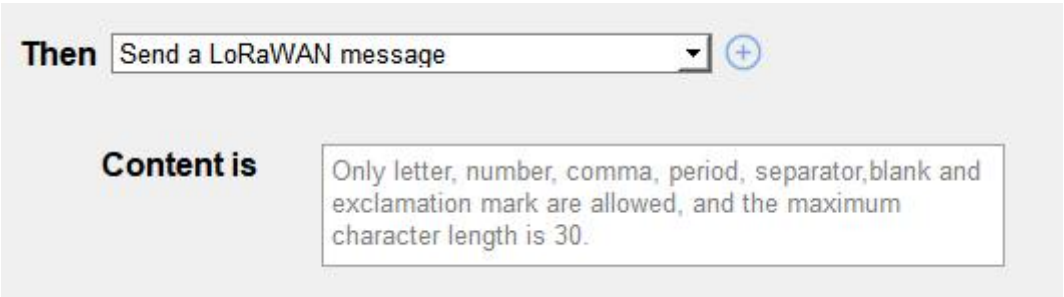
1. Go to the **Command** page, and click "**Edit**" to add commands.

2. Set an IF condition based on the terminal device data or UC100 device status.




| Condition | Description |
|---|---|
| Channel | When the value of a Modbus channel, reaches the condition.<br>For coil/discrete type, the condition is False/True;<br>For other types, the condition is Above/Below/Within/Change.<br>**Is continued for**: the threshold value should last for some time.<br>**Set lockout time:** after the lockout time, UC100 will check if the value still reaches the threshold and matches the condition. 0 means this condition will only be detected once.<br>**The time interval of value change:** the change value should last for some time.<br>**Note:** The parameter will be hidden if you enable **Two-way Pass-through** feature. |
| Received a Milesight D2D control command | This only works with the Milesight D2D feature is enabled. |

3. Set THEN action according to your request. You can add at most 3 actions in one command.

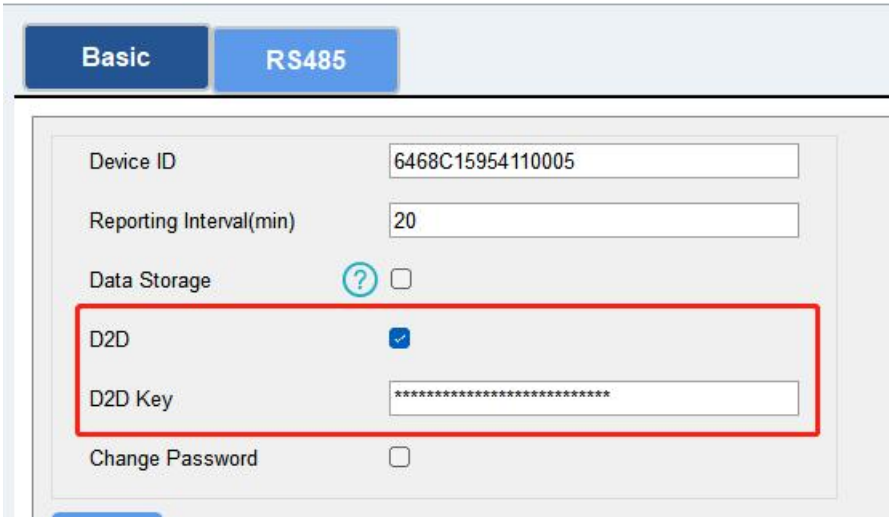


| Action | Description |
|---|---|
| Report a threshold packet | Report as threshold alarm packet to network server when the value of selected Modbus channel value reaches the threshold. |
| Report a packet on shift change | Report as change alarm packet to network server when the value of selected Modbus channel value changes a specific range. |
| Send a LoRaWAN® message | Send a custom message to the network server. |
| Restart the Device | Reboot the device. |
| Send a Milesight D2D control command | This only works with Milesight D2D feature is enabled. |
| Send a Modbus command via the RS485 interface | Send a Modbus RTU command to the RS485 device. |

## 4.6 Milesight D2D Settings

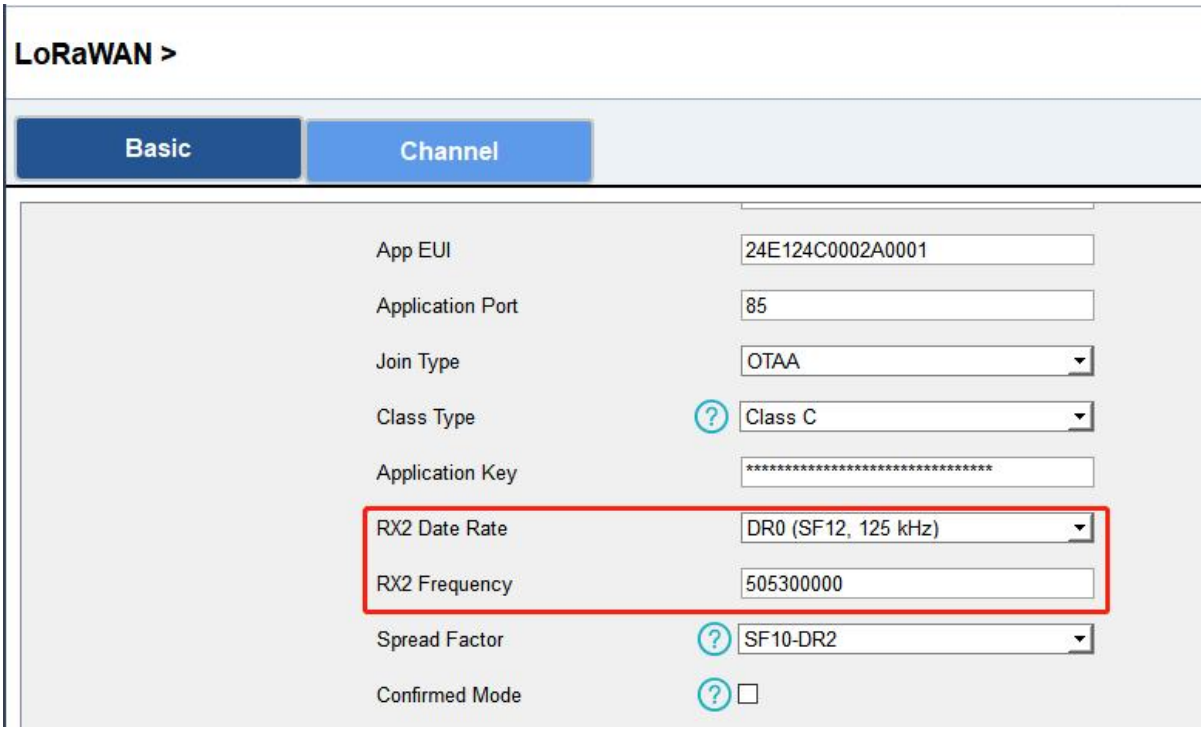
Milesight D2D protocol is developed by Milesight and used for setting up transmission among Milesight devices without a gateway. When the Milesight D2D setting is enabled, UC100 can work as a Milesight D2D controller to send control commands to other devices or work as a Milesight D2D agent to receive commands to trigger a reboot or message to the network server.

1. Go to **General > Basic** page, enable Milesight D2D feature and define a unique Milesight D2D key which is the same as Milesight D2D controller or agent devices. (Default Milesight D2D Key: 5572404C696E6B4C6F52613230313823)



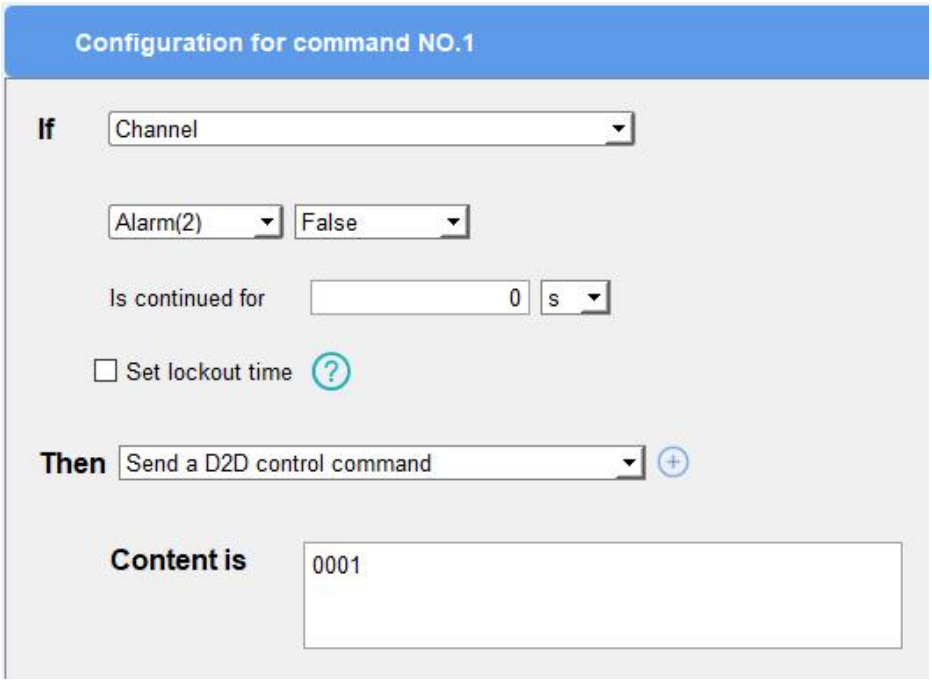

2. Go to **LoRaWAN Settings > Basic** to configure the RX2 datarate and RX2 frequency. When

UC100 works as Milesight D2D controller, it will send commands as RX2 settings.

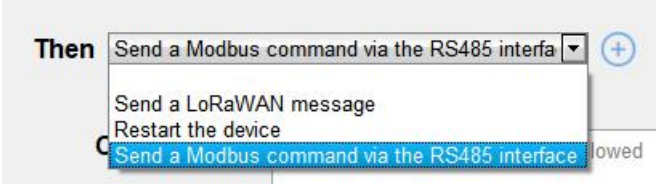


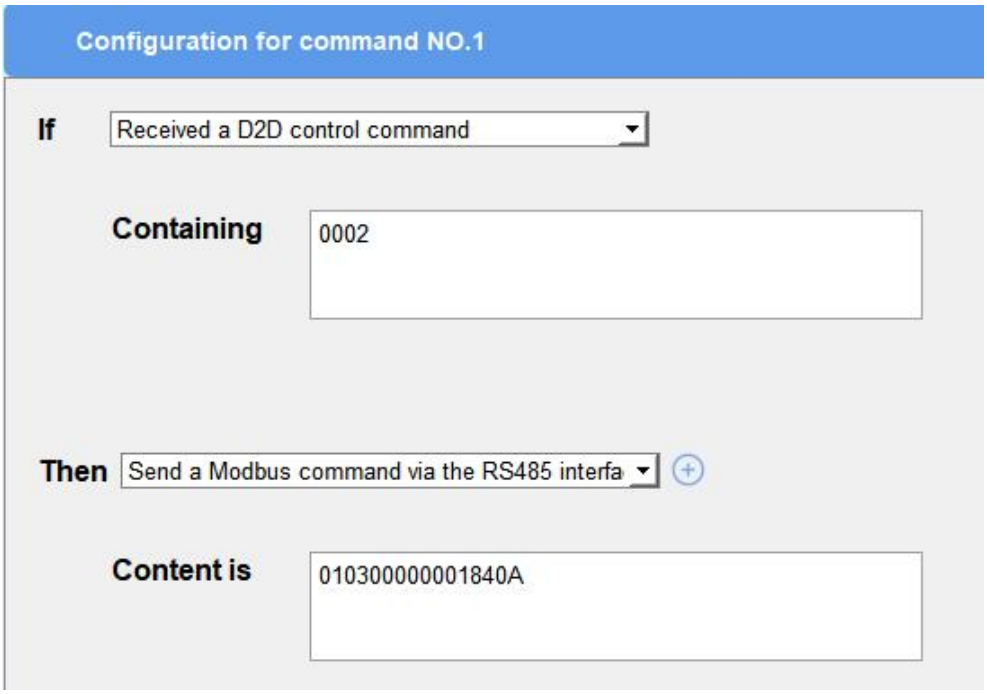

3. Go to **Command** page to set corresponding operations.

When the RS485 channel triggers, UC100 can work as Milesight D2D controller to send a control command to control the Milesight D2D agent device. The command should be a 2-byte hexadecimal number.




When UC100 receives a Milesight D2D command, it can work as a Milesight D2D agent to reboot the device, send a LoRaWAN® message or send Modbus command to RS485 terminal devices.



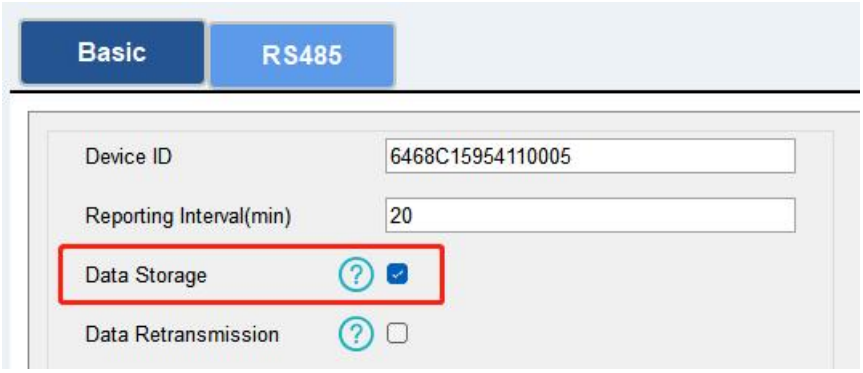




## 4.7 Data Storage

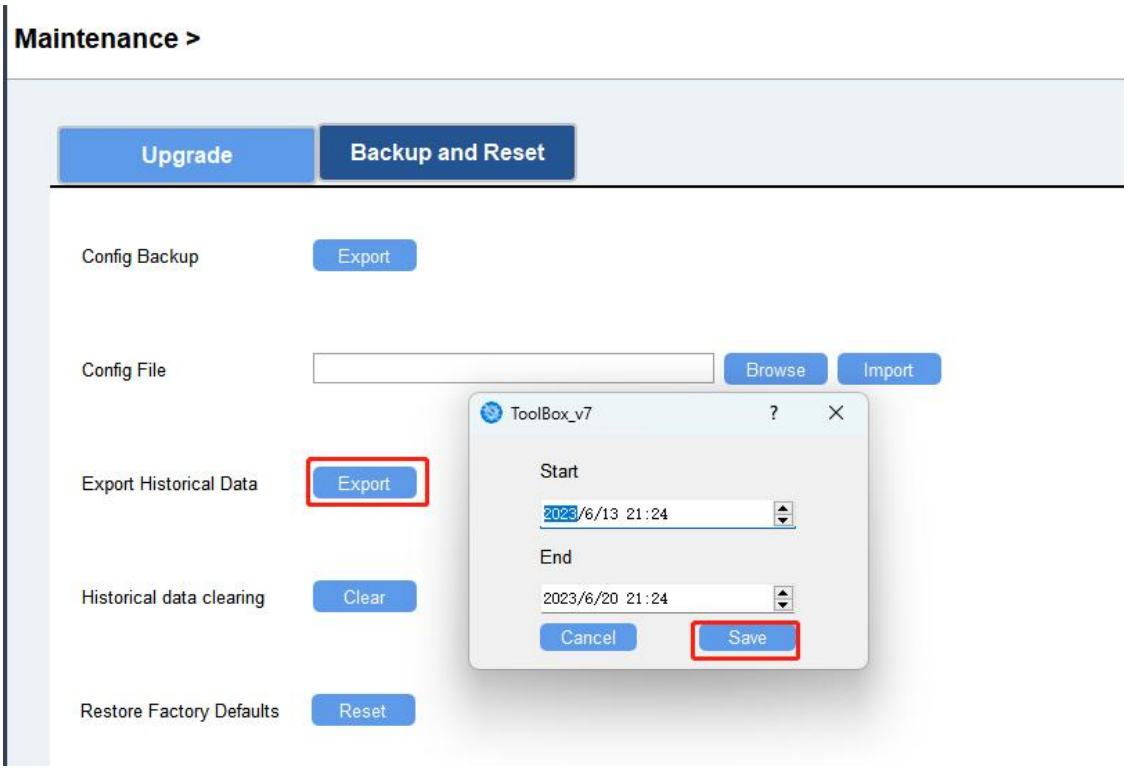
UC100 supports storing 1000 pieces of data locally and export data via ToolBox. The device will record the data according to reporting interval even not joining to network.

1. Go to **Status** page to click **Sync** to sync the device time or select LoRaWAN® version as 1.0.3 to enquire the time from network server when joining network.

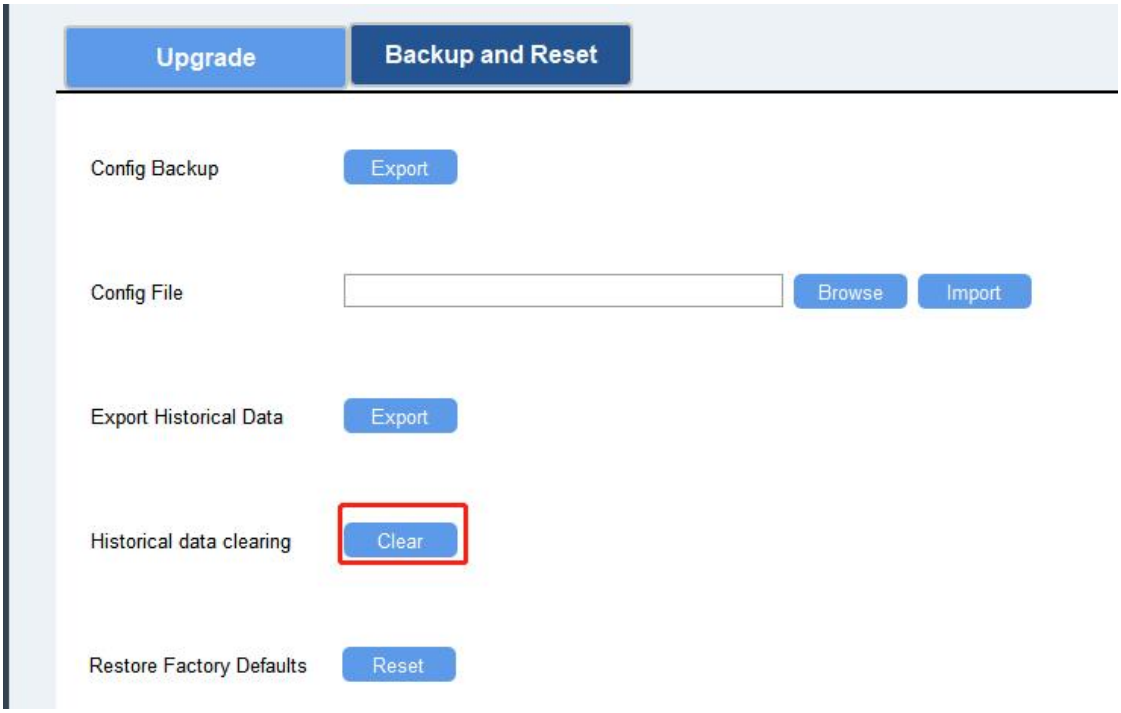

2. Go to **General > Basic** to enable data storage feature.




3. Go to **Maintenance > Backup and Reset** , click **Export** to select the data time period and click **Save** to export data.




4. Click **Clear** to clear all stored data inside the device as required.



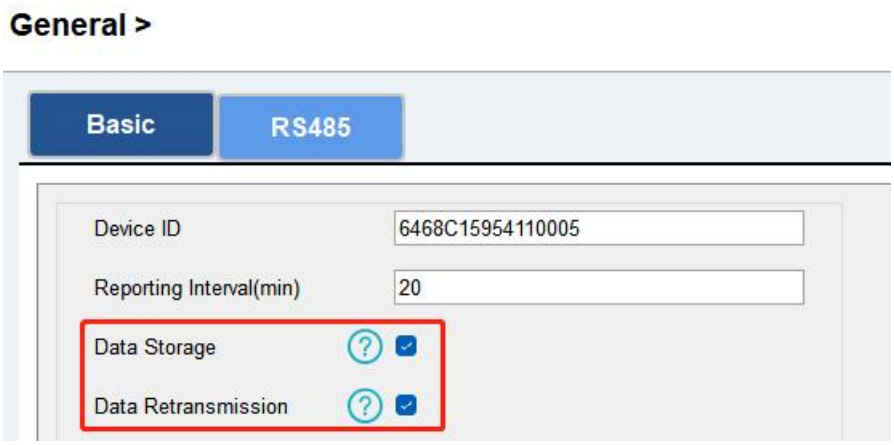

## 4.8 Data Retransmission

UC100 supports data retransmission to ensure the network server can get all data even if network is down for some times. There are two ways to get the lost data:

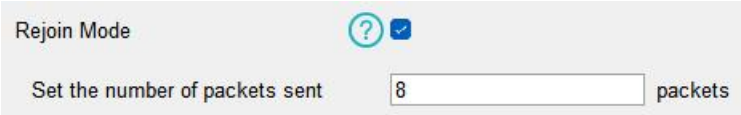
- Network server sends downlink commands to enquire the historical data for specifying time range, refer to Historical Data Enquiry;
- When network is down if no response from LinkCheckReq MAC packets for a period of time, the device will record the network disconnected time and re-transmit the lost data after device re-connects the network.

Here are the steps for retransmission:

1. Go to **General > Basic** to enable data storage and data retransmission feature.



2. Go to **LoRaWAN Settings > Basic** to enable rejoin mode feature and set the number of packets sent. Take below as example, the device will send LinkCheckReq MAC packets to the network server regularly to check if the network is disconnected; if there is no response for 8+1 times, the join status will change to de-active and the device will record a data lost time point(the time to join the network).




3. After the network connected back, the device will send the lost data from the point in time when the data was lost according to the data re-transmission reporting interval.

**Note:**

1) If the device is reboot or re-powered during data retransmission, the device will re-send interrupted retransmission data again after device is reconnected back to the network.

2) If the network is disconnected again during data retransmission, it will only send the latest disconnection data.

3) UC100 supports to send downlink commands to enquire the historical data for specifying time range, refer to Historical Data Enquiry.

## 4.9 Maintenance

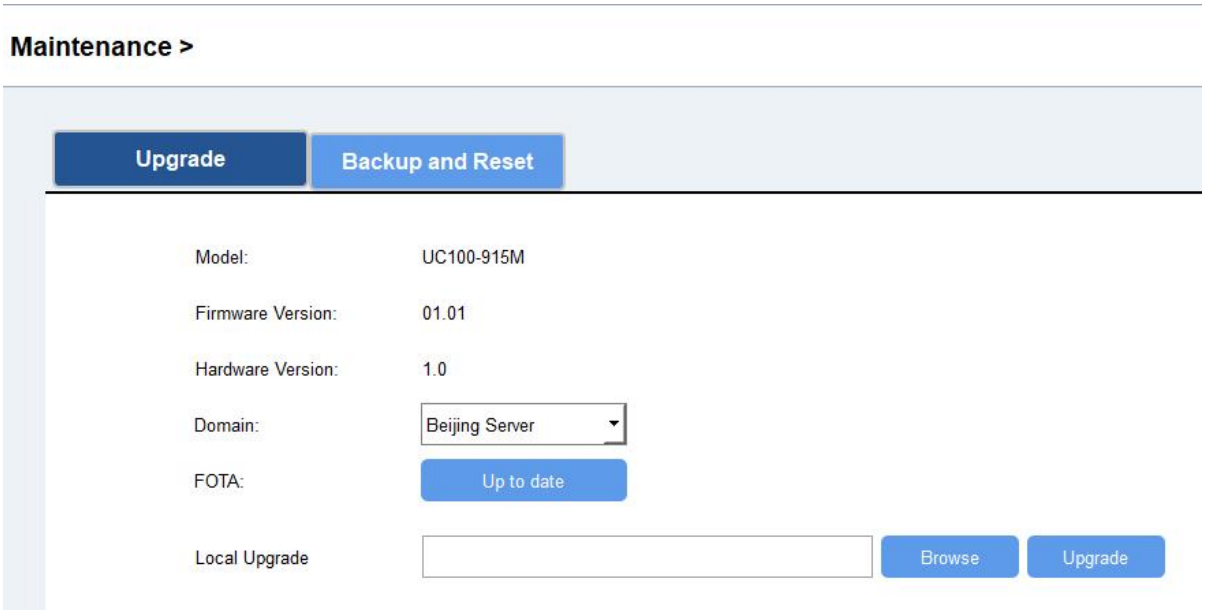
### 4.9.1 Upgrade

UC100 supports upgrade firmware locally via ToolBox software.

1. Download firmware from Milesight website to your PC.

2. Go to **Maintenance > Upgrade**, click **Browse** to import firmware and upgrade the device.

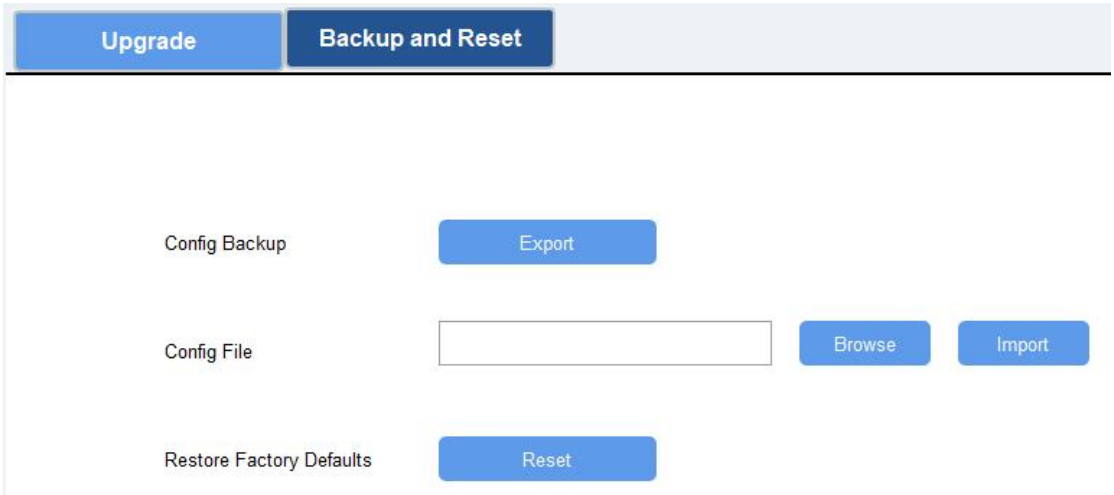
**Note:** Any operation on ToolBox is not allowed during upgrading, otherwise the upgrading will be interrupted, or even the device will break down.




### 4.9.2 Backup

UC100 devices support configuration backup for easy and quick device configuration in bulk. Backup is allowed only for devices with the same model and LoRaWAN® frequency band.

1. Go to **Maintenance > Backup and Reset**, and click **Export** to save the current configuration as json format backup file.
2. Click **Browse** to select the backup file, then click **Import** to import the configurations.

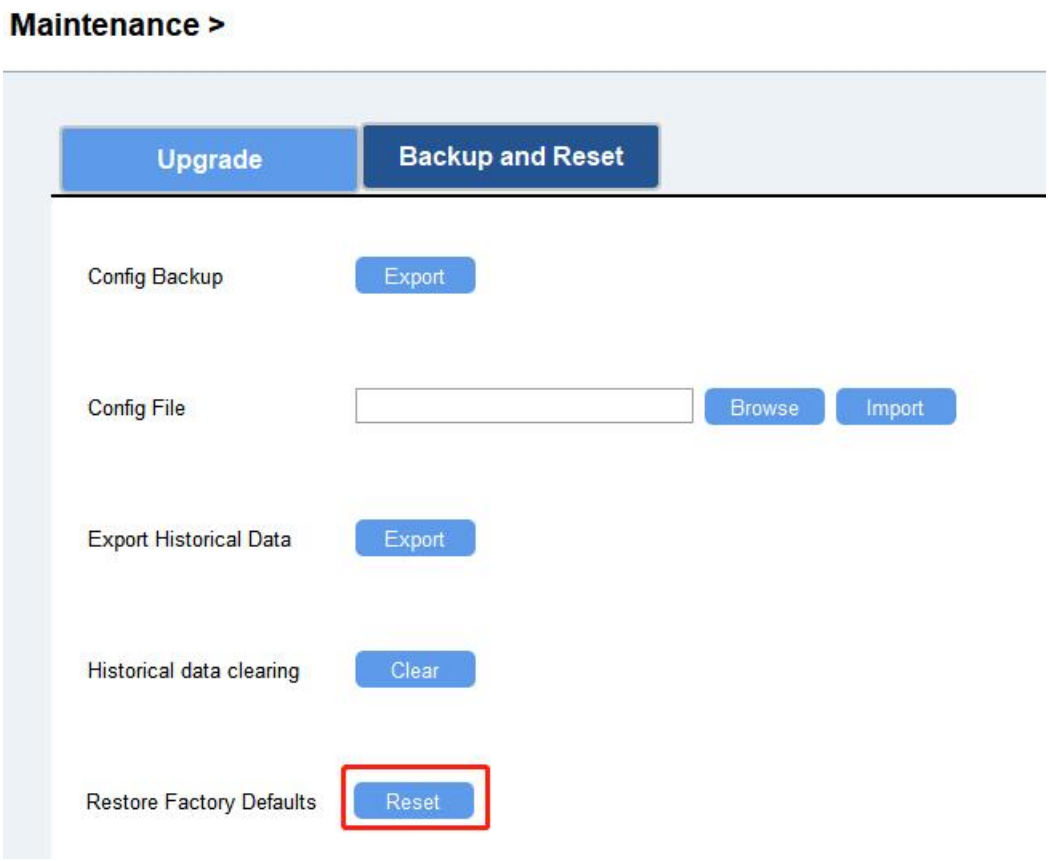



### 4.9.3 Reset to Factory Default

Please select one of following methods to reset device:

**Via Hardware:** Open the case of UC100, and hold the reset button for more than 10s until the LED blinks.

**Via ToolBox Software:** Go to **Maintenance > Backup and Reset** to click **Reset**.




## 5. Communication Protocol

All data are based on the following format (HEX), the Data field should follow little-endian:

| Channel1 | Type1 | Data1 | Channel2 | Type2 | Data2 | Channel 3 | ... |
|---|---|---|---|---|---|---|---|
| 1 Byte | 1 Byte | N Bytes | 1 Byte | 1 Byte | M Bytes | 1 Byte | ... |

For decoder examples, you can find them at https://github.com/Milesight-IoT/SensorDecoders.

### 5.1 Device Information

UC100 reports basic device information of device every time joining the network.

| Item | Channel | Type | Byte | Description |
|---|---|---|---|---|
| Protocol Version | ff | 01 | 1 | 01 => V1 |
| Hardware Version | | 09 | 2 | 01 20 => V1.2 |
| Firmware Version | | 0a | 2 | 01 01 => V1.1 |
| Power On | | 0b | 1 | Device is on |
| Device SN | | 16 | 8 | 16 digits |
| Device Type | | 0f | 1 | 00: Class A, 01: Class B, 02: Class C |
| TSL Version | | ff | 2 | 02 01 =>V2.1 |

| | | | | |
|---|---|---|---|---|
| Reset Event | | fe | 1 | ff, only report when the device resets |

**Example:**

| ff0bff ff0101 ffff0201 ff166445b43411300001 ff090100 ff0a0101 ff0f02 | | |
|---|---|---|
| **Channel** | **Type** | **Value** |
| ff | 0b (Power On) | ff |
| ff | 01 (Protocol Version) | 01 (V1) |
| ff | ff (TSL version) | 02 01=>V2.1 |
| ff | 16 (Device SN) | 64 45 B4 34 11 30 00 01 |
| ff | 09 (Hardware Version) | 0100 (V1.0) |
| ff | 0a (Firmware Version) | 0101 (V1.1) |
| ff | 0f (Device Type) | 02=Class C |

## 5.2 Modbus Channel Data

UC100 reports RS485 sensor data which are fetched by Modbus channels according to reporting interval (20 mins by default).

<table>
<tr><th>Item</th><th>Channel</th><th>Type</th><th>Byte</th><th>Description</th></tr>
<tr><td>Modbus Channel</td><td>ff</td><td>19</td><td>4~7</td><td>Channel ID (1B) + Data Length (1B) + Data Type (1B) +Data (Mutable)<br>Data Type:<br>Bit 7: 0 = unsigned, 1 = signed<br>Bit 6-0:
<table>
<tr><th>Code</th><th>Data Type</th></tr>
<tr><td>0000000</td><td>Coil</td></tr>
<tr><td>0000001</td><td>Discrete</td></tr>
<tr><td>0000010</td><td>Input16</td></tr>
<tr><td>0000011</td><td>Hold16</td></tr>
<tr><td>0000100</td><td>Hold32</td></tr>
<tr><td>0000101</td><td>Hold_float</td></tr>
<tr><td>0000110</td><td>Input32</td></tr>
<tr><td>0000111</td><td>Input_float</td></tr>
<tr><td>0001000</td><td>Input_int32_with upper 16 bits</td></tr>
<tr><td>0001001</td><td>Input_int32_with lower 16 bits</td></tr>
<tr><td>0001010</td><td>Hold_int32_with upper 16 bits</td></tr>
<tr><td>0001011</td><td>Hold_int32_with lower 16 bits</td></tr>
</table></td></tr>
<tr><td>Collection Exception</td><td>ff</td><td>15</td><td>1</td><td>Channel ID of failed Modbus collection.</td></tr>
</table>

**Note:** Channel ID can be configured in ToolBox.

| Channel ID | Description |
|---|---|
| 00 | RS485 (Modbus Master) Channel 1 |
| 01 | RS485 (Modbus Master) Channel 2 |
| 02 | RS485 (Modbus Master) Channel 3 |
| ... | ... |
| 1f | RS485 (Modbus Master) Channel 32 |

## Examples:

1. No Modbus Channel.

| ff0bff | | |
|---|---|---|
| **Channel** | **Type** | **Value** |
| ff | 0b (Power On) | ff |

2. Fail to fetch Channel1 data.

| ff 15 00 | | |
|---|---|---|
| **Channel** | **Type** | **Value** |
| ff | 15 | 00 => Channel 1 |

3. Succeed to fetch Channel8 data.

| ff 19 07 02 03 15 00 | | | | | |
|---|---|---|---|---|---|
| **Channel** | **Type** | **Channel ID** | **Data Size** | **Data Type** | **Value** |
| ff | 19 | 07 => Channel 8 | 02 => 2 bytes | 03 => Unsigned Hold 16 | 15 00 => 00 15 = 21 |

**Note:** When data type is holding register or input register, ToolBox can set different byte orders. Take below Modbus register response from RS485 sensors as example:

| Register Address | Value (Hex) |
|---|---|
| 0 | 00 15 |
| 1 | 00 20 |

When using different byte orders, you can use ToolBox to fetch different results, and the device will upload data with little endian order.

| Data Type | Byte Order | Fetch Result | Uplink (HEX) |
|---|---|---|---|
| Holding/Input Register (INT16) | AB | 21 (0x15) | 15 00 (BA) |
| | BA | 5376 (0x1500) | 00 15 (AB) |
| Holding/Input Register (INT32) | ABCD | 1376288 (0x00150020) | 20 00 15 00 (DCBA) |
| | CDAB | 2097173 (0x00200015) | 15 00 20 00 (BADC) |
| | BADC | 352329728 | 00 20 00 15 |

| | | | |
|---|---|---|---|
| | | (0x15002000) | (CDAB) |
| | DCBA | 536876288 (0x20001500) | 00 15 00 20 (ABCD) |
| Holding/Input Register (INT32 with upper 16 bits) | / | 21 (0x15) | 15 00 00 00 |
| Holding/Input Register (INT32 with lower 16 bits) | / | 32 (0x20) | 20 00 00 00 |

## 5.3 Modbus Channel Alarm

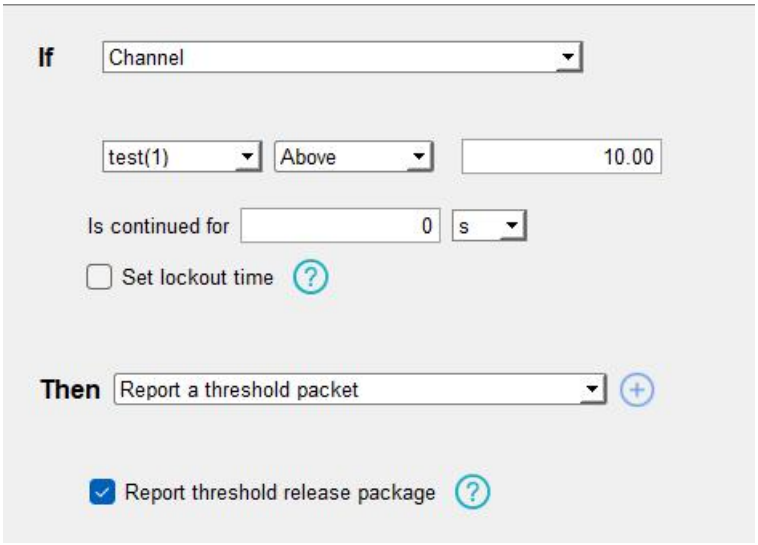
UC100 supports to report Modbus channel alarm packets if a Modbus channel value reaches the condition.

**Note:** when data type is coil or discrete, the device will not report the alarm packet.

| Item | Channel | Type | Byte | Description |
|---|---|---|---|---|
| Modbus Channel Alarm | ff | ee | 4~7 | **Note: this only works when firmware version is 1.7 or later.**<br>Alarm Type (1B) + Data Length (1B) + Data Type (1B) +Data (Mutable)<br>**Alarm Type:**<br>Bit7-6: 00=No, 01=Threshold alarm, 10=Threshold Alarm release, 11=Change alarm<br>Bit 5-0: Channel ID, range: 0-31 |
| Modbus Channel Change Alarm | f9 | 5f | 7 | **Note: this only works when firmware version is 1.9 or higher.**<br>Alarm Type (1B) + Data Length (1B) + Data Type (1B) +Change Value (4B, float32)<br>**Alarm Type:**<br>Bit7-6: 11=Change alarm<br>Bit 5-0: Channel ID, range: 0-31 |

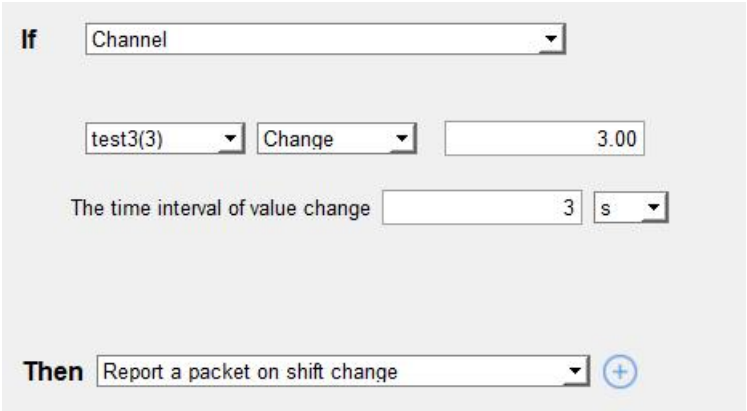
**Examples:**

1. Channel 1 data reaches the threshold.




| f95f 40 02 03 1500 00000000 | | | | | | |
|---|---|---|---|---|---|---|
| **Channel** | **Type** | **Channel ID & Alarm Type** | **Data Size** | **Data Type** | **Value** | **Change Value** |
| f9 | 5f | 40 => 0100 0000<br>01=Threshold alarm<br>100000=<br>00 => Channel 1 | 02 =><br>2 bytes | 03 =><br>Unsigned<br>Hold 16 | 15 00 =><br>00 15 = 21 | 00000000 |

2. Channel 3 data changes more than 3.




| ff ee c2 02 03 05 00 f95f c2 02 03 00004040 | | | | | |
|---|---|---|---|---|---|
| **Channel** | **Type** | **Channel ID & Alarm Type** | **Data Size** | **Data Type** | **Value** |
| ff | ee | C2 => 1100 0010<br>11=Change alarm<br>000010=<br>02 => Channel 3 | 02 =><br>2 bytes | 03 => Hold 16 | 05 00 =><br>00 5 = 5 |
| **Channel** | **Type** | **Channel ID & Alarm Type** | **Data Size** | **Data Type** | **Change Value** |

| | | | | | |
|---|---|---|---|---|---|
| f9 | 5f | c2 => 1100 0010<br>11=Change alarm<br>000010=<br>02 => Channel 3 | 02 =><br>2 bytes | 03 =><br>Unsigned<br>Hold 16 | 00 00 40<br>40=>40 40 00<br>00=3 |

## 5.4 Downlink Command

UC100 supports downlink commands to configure the device. The application port is 85 by default.

<table>
<tr><th>Item</th><th>Channel</th><th>Type</th><th>Description</th></tr>
<tr><td>Reporting Interval</td><td rowspan="9">ff</td><td>03</td><td>2 Bytes, unit: s</td></tr>
<tr><td>Reboot</td><td>10</td><td>ff</td></tr>
<tr><td>Data Storage</td><td>68</td><td>00: disable, 01: enable</td></tr>
<tr><td>Data Retransmission</td><td>69</td><td>00: disable, 01: enable</td></tr>
<tr><td>Data Retransmission Interval</td><td>6a</td><td>3 Bytes<br>Byte 1: 00<br>Byte 2-3: interval time, unit:s<br>range: 30~1200s (600s by default)</td></tr>
<tr><td>Modbus Channel Setting</td><td>ef</td><td>01+Channel ID (1B)+Slave ID (1B) + Address (2B) + Type (1B) + Sign (1B)<br>Sign: 11=signed, 01=unsigned<br>Type:
<table>
<tr><th>Code</th><th>Data Type</th></tr>
<tr><td>00</td><td>Coil</td></tr>
<tr><td>01</td><td>Discrete</td></tr>
<tr><td>02</td><td>Input16_AB</td></tr>
<tr><td>03</td><td>Input16_BA</td></tr>
<tr><td>04</td><td>Input32_ABCD</td></tr>
<tr><td>05</td><td>Input32_BADC</td></tr>
<tr><td>06</td><td>Input32_CDAB</td></tr>
<tr><td>07</td><td>Input32_DCBA</td></tr>
<tr><td>08</td><td>Input32_AB</td></tr>
<tr><td>09</td><td>Input32_CD</td></tr>
<tr><td>0a</td><td>Input_float_ABCD</td></tr>
<tr><td>0b</td><td>Input_float_BADC</td></tr>
<tr><td>0c</td><td>Input_float_CDAB</td></tr>
<tr><td>0d</td><td>Input_float_DCBA</td></tr>
<tr><td>0e</td><td>Hold16_AB</td></tr>
</table></td></tr>
</table>

| | | | | |
|---|---|---|---|---|
| | | | 0f | Hold16_BA |
| | | | 10 | Hold32_ABCD |
| | | | 11 | Hold32_BADC |
| | | | 12 | Hold32_CDAB |
| | | | 13 | Hold32_DCBA |
| | | | 14 | Hold32_AB |
| | | | 15 | Hold32_CD |
| | | | 16 | Hold_float_ABCD |
| | | | 17 | Hold_float_BADC |
| | | | 18 | Hold_float_CDAB |
| | | | 19 | Hold_float_DCBA |
| Delete Modbus Channel | | ef | 00+Channel ID (1B) | |
| Mobus Channel Name | | ef | 02+Channel ID (1B) + Name Length (1B) + Name (Mutable) | |

**Note:** Channel ID in downlink commands is different from uplinks:

| Channel ID | Description |
|---|---|
| 01 | RS485 (Modbus Master) Channel 1 |
| 02 | RS485 (Modbus Master) Channel 2 |
| 03 | RS485 (Modbus Master) Channel 3 |
| ... | ... |
| 20 | RS485 (Modbus Master) Channel 32 |

**Examples:**

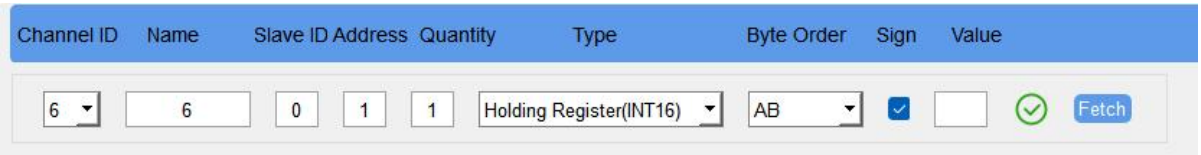
1. Set reporting interval as 20 minutes.

| ff 03 b0 04 | | |
|---|---|---|
| **Channel** | **Type** | **Value** |
| ff | 03 | b0 04 => 04 b0 = 1200 s = 20 mins |

2. Reboot the device

| ff 10 ff | | |
|---|---|---|
| **Channel** | **Type** | **Reserved** |
| ff | 10 (Reboot) | ff |

3. Add a Modbus channel as below:




| ff ef 01 06 00 0100 0e 11 | | |
|---|---|---|
| **Channel** | **Type** | **Value** |
| ff | ef | Channel: 06=Channel 6 |

| | | |
|---|---|---|
| | | Slave ID: 00=Slave ID<br>Address: 01 00=>00 01=1<br>Type: 0e=Hold16_AB<br>Sign: 11=signed |

4. Set name of Modbus channel6 as “test6”.

| ff ef 02 06 05 7465737436 | | |
|---|---|---|
| **Channel** | **Type** | **Value** |
| ff | ef | Channel: 06=Channel 6<br>Name length: 05=5 Bytes<br>Hex to ASCii: 74 65 73 74 36 => t e s t 6 |

## 5.5 Historical Data Enquiry

UC100 supports sending downlink commands to enquire historical data for specified time point or time range. Before that, ensure the device time is correct and data storage feature was enabled to store the data.

**Command format:**

| Channel | Type | Description |
|---|---|---|
| fd | 6b (Enquire data in time point) | 4 Bytes, unix timestamp |
| fd | 6c (Enquire data in time range) | Start time (4 bytes) + End time (4 bytes), Unix timestamp |
| fd | 6d (Stop query data report) | ff |
| ff | 6a (Report Interval) | 3 Bytes<br>Byte 1: 01<br>Byte 2-3: interval time, unit:s<br>range: 30~1200s (60s by default) |

**Reply format:**

| Channel | Type | Description |
|---|---|---|
| fc | 6b/6c | 00: data enquiry success<br>01: time point or time range invalid<br>02: no data in this time or time range |
| 20 | ce (Modbus Channel) | Data time stamp (4B) + Channel ID (1B) + Ctrl (1B) + Data (4B) |
| 20 | cd (Custom Message) | Data time stamp (4B) + Data Length (1B) + Data (Mutable) |

Ctrl Format:

| Bit | 7 | 6-2 | 1 | 0 |
|---|---|---|---|---|
| | 0: unsigned<br>1: signed | Data Type | 0: fetch failure<br>1: fetch success | 0 |

**Note:**

1. The device only uploads no more than 300 data records per range enquiry.
2. When enquiring the data in time point, it will upload the data which is closest to the search point within the reporting interval range. For example, if the device reporting interval is 10 minutes and users send command to search for 17:00's data, if the device find there is data stored in 17:00, it will upload this data; if not, it will search for data between 16:50 to 17:10 and upload the data which is closest to 17:00.

**Example:**

1. Enquire historical data between 2024/01/18 15:45:00 to 2024/01/18 15:50:00.

| fd6c fcd6a865 28d8a865 | | |
|---|---|---|
| **Channel** | **Type** | **Value** |
| fd | 6c (Enquire data in time range) | Start time: fcd6a865 => 65a8d6fc = 1705563900 =2024/01/18 15:45:00<br>End time: 28d8a865 => 65a8d828 = 1705564200 =2024/01/18 15:50:00 |

Reply:

| fc6c00 | | |
|---|---|---|
| **Channel** | **Type** | **Value** |
| fc | 6c (Enquire data in time range) | 00: data enquiry success |

| 20ce 93d7a865 00 3a 15000000 | | | |
|---|---|---|---|
| **Channel** | **Type** | **Time Stamp** | **Value** |
| 20 | ce (Modbus Channel) | 93d7a865 => 2024/01/18 15:47:00 | 00: Channel 1<br>Crtl: 3a => 0011 1010<br>Bit1=1=> Fetch success<br>Bit7-2 => 001110 = 0e=Unsigned Hold16<br>Data: 15000000=>00 00 00 15=21 |

**-END-**