Milesight

IoT Controller

UC100

User Guide

Contents

Chapter 1. Introduction

Copyright Statement

Safety Instruction

These instructions are intended to ensure that user can use the product correctly to avoid danger or property loss. Milesight will not shoulder responsibility for any loss or damage resulting from not following the instructions of this operating guide.

CAUTION:

Injury or equipment damage may be caused if any of these cautions are neglected.

- The device must not be disassembled or 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.

Services

Milesight provides customers with timely and comprehensive technical support services. End-users can contact your local dealer to obtain technical support. Distributors and resellers can contact directly with Milesight for technical support.

Technical Support Mailbox: iot.support@milesight.com

Online Support Portal: https://support.milesight-iot.com

Resource Download Center: https://www.milesight.com/iot/resources/download-center/

MILESIGHT CHINA

TEL: +86-592-5085280

FAX: +86-592-5023065

Add: Building C09, Software Park Phase III, Xiamen 361024, Fujian, China

Revision History

Release Date	Version	Revision Content
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 2. Increase to 32 Modbus channels 3. Add downlink commands to configure Modbus channels 4. Add Modbus channel alarm feature
Jan. 7, 2025	V 1.3	1. Add optional power converter kit accessory 2. Support to report sign type of Modbus channels 3. Add Modbus channel change alarm report 4. Add device class type, TSL version and reset report
Jun. 19, 2025	V 2.0	1. Add the external antenna version 2. Modbus channel supports double64 and int64 3. Support two registers for per Modbus channel 4. Add time zone settings and daylight saving time setting 5. Add IF-THEN commands

Chapter 2. Product Introduction

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 dropped. Combining with Milesight Development Platform solution, it can monitor and control remote assets via web server easily.

UC100 is widely used in indoor applications like smart metering, smart cities, building automation, etc.

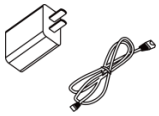
Features

- Support reading 32 Modbus registers, can connect up to 32 Modbus devices
- Ultra-wide-distance transmission up to line of sight of 15 km
- Multiple triggering conditions and actions
- Embedded watchdog for device working stability
- Store historical data locally and support retransmission to prevent data loss
- Support Milesight D2D protocol to enable ultra-low latency and directly control without LoRaWAN® gateway
- Compliant with standard LoRaWAN® gateways and network servers
- Quick and easy management with Milesight Development Platform solution
- Support Firmware Update Over the Air (FUOTA) feature

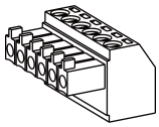
Chapter 3. Hardware Introduction

Packing List

1 × UC100 Controller



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



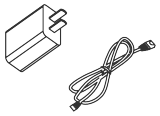
1 × Terminal Block

2 × Wall Mounting Kits

1 × LoRaWAN® Magnetic Antenna

1 × Quick Guide

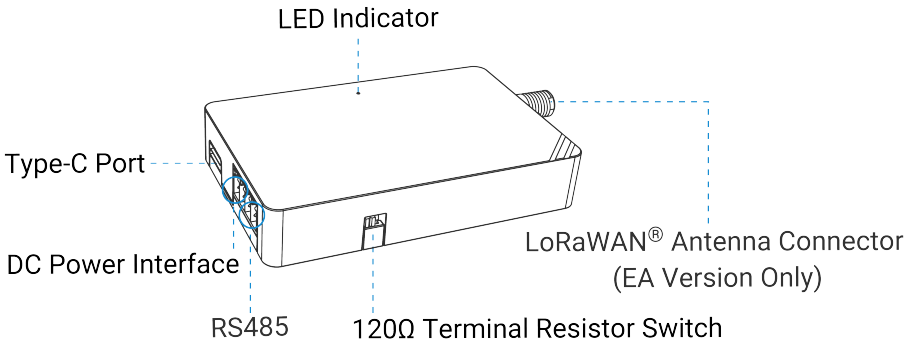
1 × Warranty Card



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

Note:
If any of the above items is missing or damaged, please contact your sales representative.

Hardware Overview



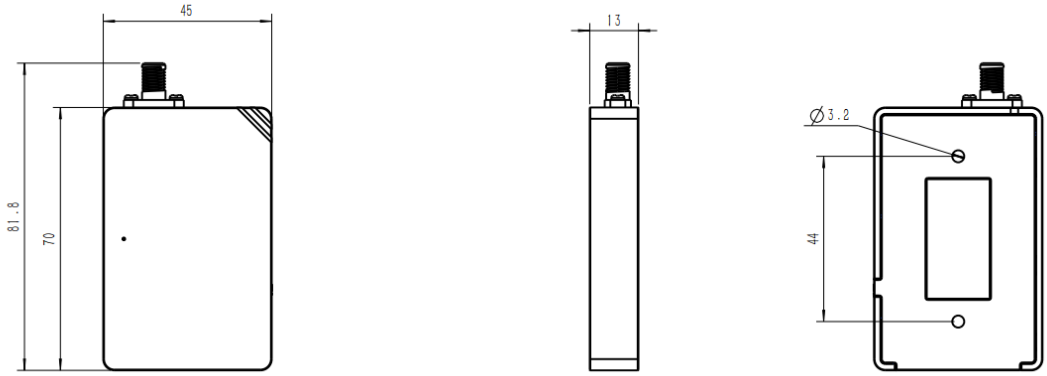
120Ω Terminal Resistor Switch: the device will add a 120Ω termination resistor to avoid data-corrupting reflections if RS485 data rate is high or cable length is long.

LED and Reset Button

The reset button is inside the device.

Function	Action	LED Indicator
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

Dimensions (mm)



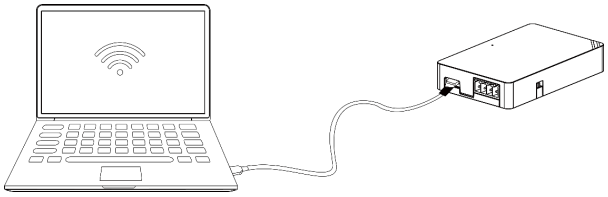
Chapter 4. Operation Guide

ToolBox Configuration Guide

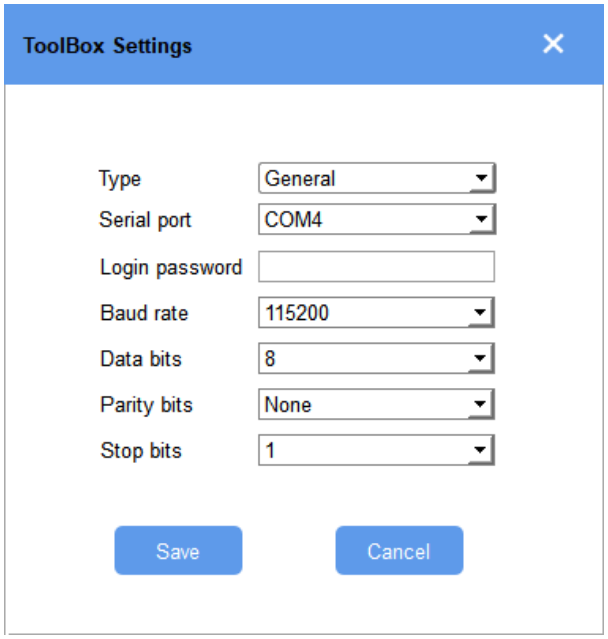
This chapter describe the steps to quickly configure this device to set up the connection with LoRaWAN® gateway and network server. If you require more advanced settings, please refer to the operation guide chapter.

1. Download ToolBox software from Milesight website.

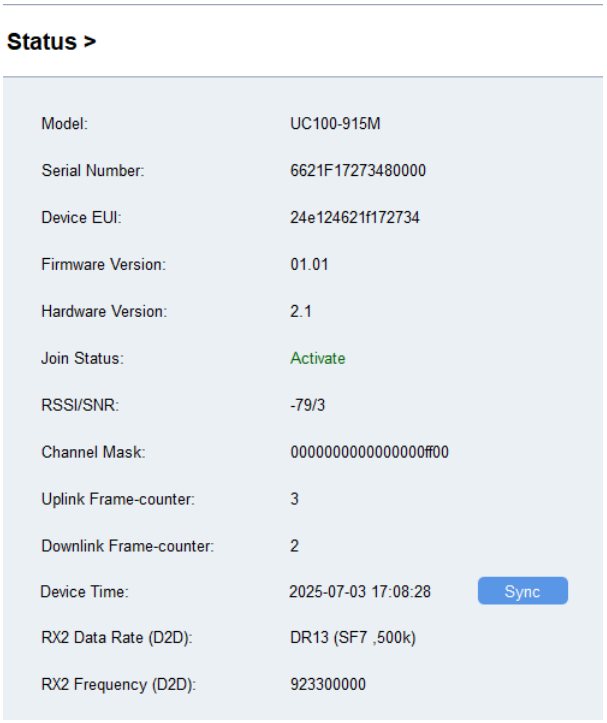
2. Power on the UC100 device, then connect it to a computer via the Type-C port.



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



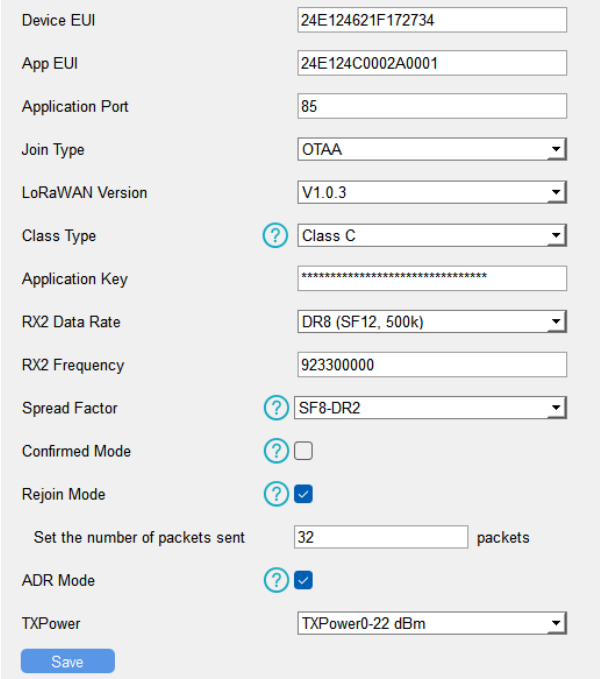
4. After logging in to the ToolBox, you can change device settings.



LoRaWAN Settings

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

Basic Settings



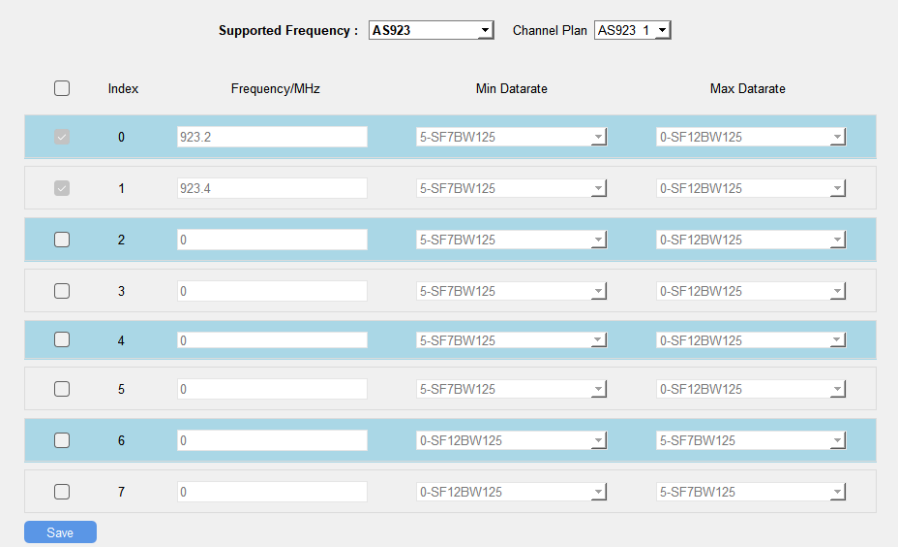
Parameter	Description
Device EUI	Unique ID of the device which can be found on the device. **Note:** please contact sales for device EUI list if you have many units.
App EUI	The default App EUI (join EUI) is 24E124C0002A0001.
Application Port	The port used for sending and receiving data, the default port is 85.
LoRaWAN® Version	V1.0.2 and V1.0.3 are available.
Class Type	Fixed as Class C.
Confirmed Mode	If the device does not receive ACK packet from network server, it will re-send data once.
Join Type	OTAA and ABP mode are available. **Note:** it's necessary to select OTAA mode if connecting device to Milesight Development Platform.

Parameter	Description
Application Key	Appkey for OTAA mode, the default is 5572404C696E6B4C6F52613230313823. **Note:** please contact sales if you require random App Keys for bulks of devices before purchase.
Network Session Key	Nwkskey for ABP mode, the default is 5572404C696E6B4C6F52613230313823.
Application Session Key	Appskey for ABP mode, the default is 5572404C696E6B4C6F52613230313823.
Device Address	DevAddr for ABP mode, default is the 5th to 12th digits of SN.
Rejoin Mode	Reporting interval≤35 mins: the device will send a specific number of Link-CheckReq MAC packets to the network server every reporting interval or every double reporting interval to validate connectivity; If there is no re-sponse, the device will re-join the network. 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. **Note:** 1. Only OTAA mode supports rejoin mode. 2. The actual sending number is **Set the number of packets sent** +1.
ADR Mode	Enable or disable network server to adjust Spreading Factor, Bandwidth an Tx Power to optimize data rates, airtime and energy consumption in the network.
Spreading Factor	If ADR mode is disabled, the device will send uplink data following this SF parameter. The higher the spreading factor, the longer the transmission

Parameter	Description
	distance, the slower the transmission speed and the more the consumption.
Tx Power	Tx power (transmit power) refers to the strength of the outgoing signal transmitted by the device. This is defined by LoRa alliance.
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

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

Examples:

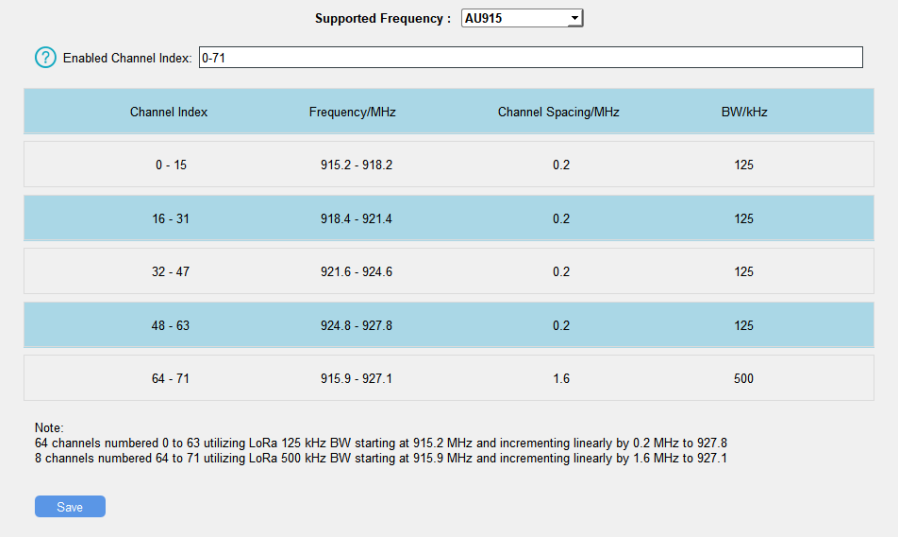
1, 40: Enabling Channel 1 and Channel 40

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: Indicate that all channels are disabled



Time Settings

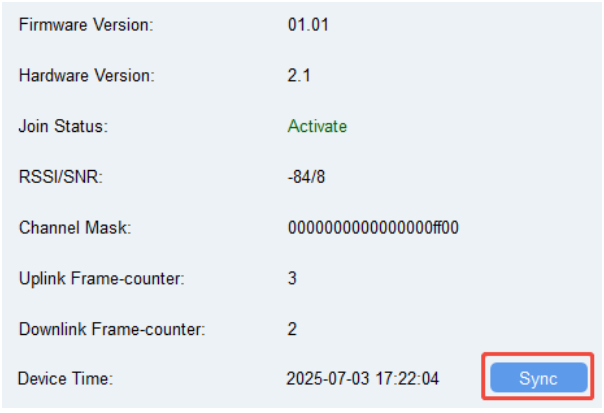
It is necessary for device to get the correct time and time zone for schedule plan and time switche controls.

Time Synchronization

Please select one of methods below to sync the time and time zone of the device.

Sync via ToolBox Software

Go to **Status** page to click **Sync** to sync the device time.



Sync via LoRaWAN® Network Server

Please ensure that the LoRaWAN® network server supports the device time synchronization feature.

1. Set the LoRaWAN® version of the device to V1.0.3.
2. Connect the device to the network server. After joining the network, the device will send a DeviceTimeReq MAC command to enquire the time from network server.

Note:
This only supports to get the time but not time zone. The time zone can be configured by ToolBox *(on page 17)* or downlink command *(on page 42)*.

Daylight Saving Time

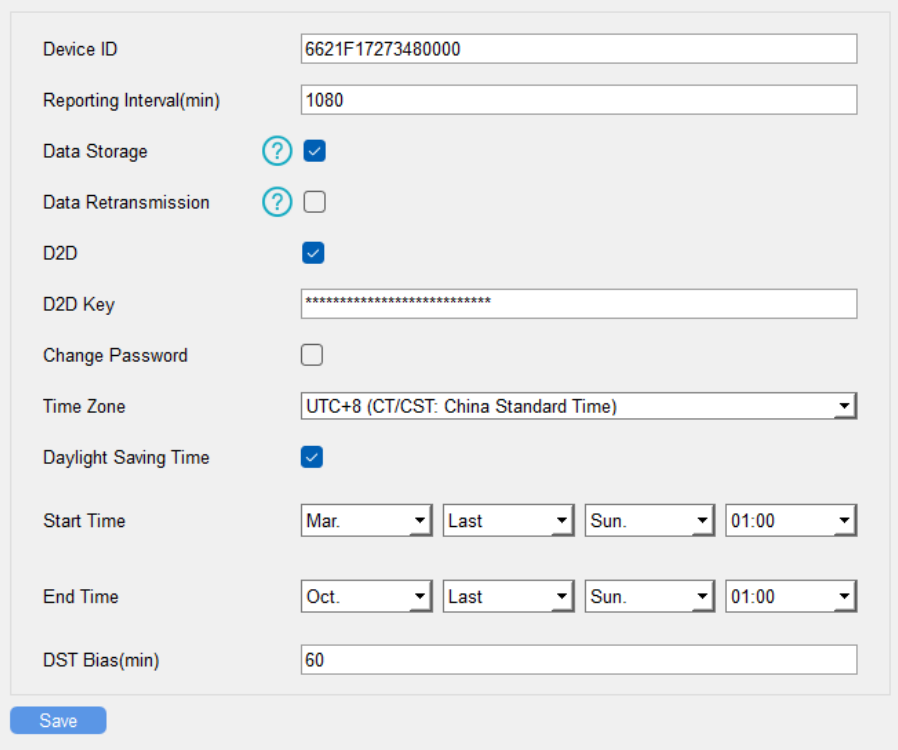
UC100 is able to configure the daylight saving time (DST) setting for accurate time control.

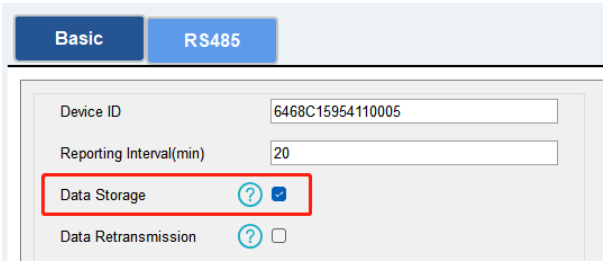
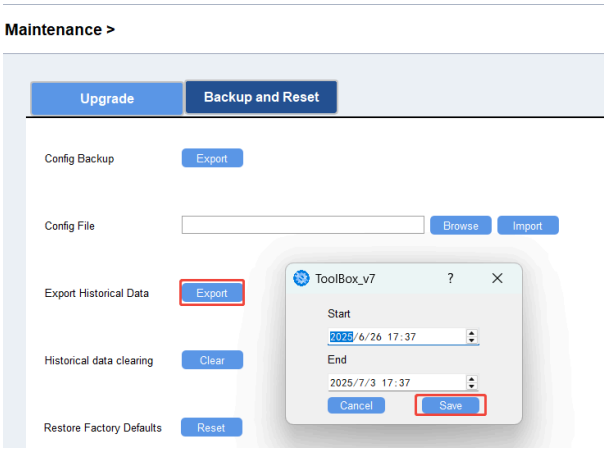
Configuration Description:

- ToolBox *(on page 17)*
- Downlink Command *(on page 42)*

General Settings

General settings include the basic parameters of the device.



Parameter	Description
Device ID	Show the SN of the device.
Reporting Interval	The interval to report transmitting Modbus channel data to network server. Range: 1-1080 minutes, Default: 20 minutes.
Data Storage	Disable or enable to store data locally. The stored data can be exported via ToolBox.   **Note:** 1. It is necessary to sync the time *(on page 14)* to ensure the data is stored in correct time. 2. The device will still store the data even the network status is de-activated.
Data Retransmission	Disable or enable data retransmission. When the device detects the network status is de-activated via Rejoin Mode *(on page 12)*, the de-

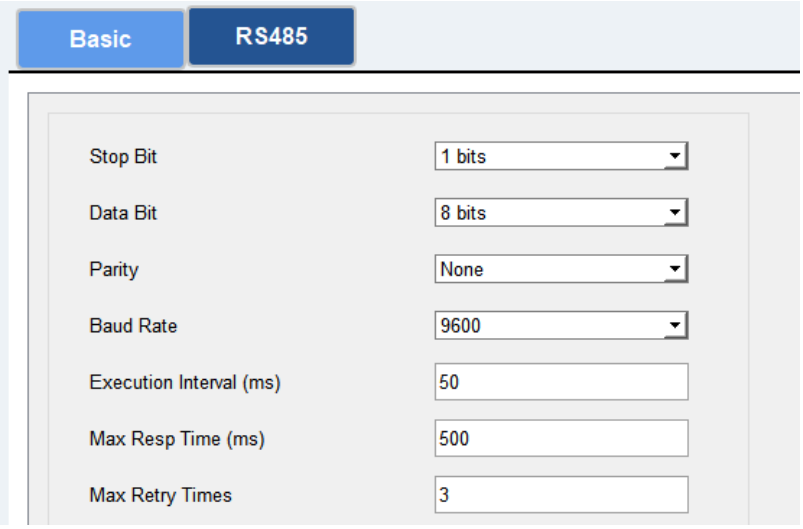
Parameter	Description
	vice will record a data lost time point and re-transmit the lost data after device re-connects to the network. **Note:** 1. This setting only takes effect when Data Storage *(on page 16)* is enabled. 2. If the device is rebooted or re-power when data retransmission is not completed, the device will re-send all retransmission data again after device is reconnected to the network. 3. If the network is disconnected again during data retransmission, it will only send the latest disconnected data. 4. The default report data retransmission interval is 600s, this can be changed via downlink command. 5. The reported format of retransmission data will include timestamps and is different from periodic report data.
D2D	Enable or disable Milesight D2D *(on page 25)* 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 device password for ToolBox software to write this device.
Time Zone	Select the UTC time zone.
Daylight Saving Time	Enable or disable Daylight Saving Time (DST). **Start Time:** the start time of DST time range. **End Time:** the end time of DST time range. **DST Bias:** the DST time will be faster according to this bias.

RS485 Settings

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

Basic Serial Settings

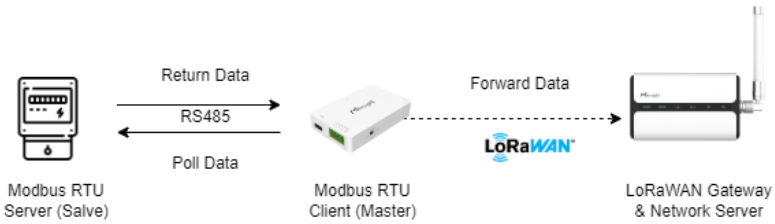
UC100 has one RS485 port for Modbus RTU device connection. The basic serial settings should be the same as RS485 terminal devices.



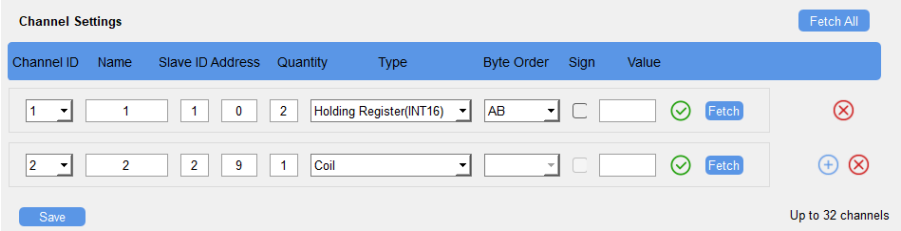
Parameter	Description
Stop Bit	1 bit/2 bit is available.
Data Bit	8 bit is available.
Parity	None, Odd and Even are available.
Baud Rate	1200/2400/4800/9600/19200/38400/57600/115200 are available.
Execution Interval (ms)	The execution interval between each Modbus channel command.
Max Resp Time (ms)	The 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

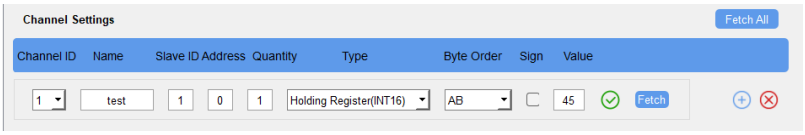
UC100 supports to work as a Modbus RTU Client (Master) to poll the data from the RS485 device and return the data to the network server.



Click ⊕ to add Modbus channels, then save configurations.



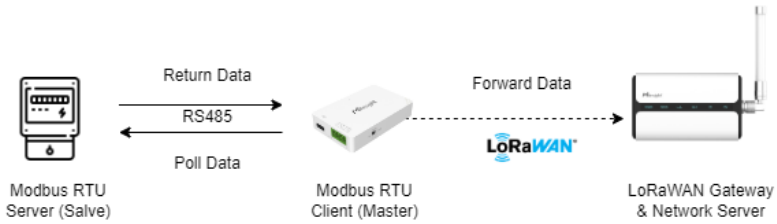
Parameter	Description
Channel ID	Select the channel ID you want to configure from 32 channels.
Name	Customize the name to identify every Modbus channel.
Slave ID	Set a Modbus slave ID of a terminal device.
Address	The starting address for reading.
Quantity	Set the number of registers to read. It can be configured to 1 or 2.
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. INT64/Double: ABCDEFGH, GHEFCDBA, BADCFEHG, HGFEDCBA INT32/Float: ABCD, CDBA, BADC, DCBA INT16: AB, BA
Sign	The tick indicates that the value has a plus or minus sign.

Parameter	Description
Fetch	Click to send a Modbus read command to test if the RS485 device can reply with the correct values. **Note:** Do not click frequently to avoid the fetch failure due to the slow response of RS485 devices. **Example:** 

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:

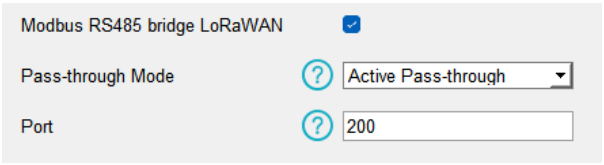
Active Pass-through: the network server can send any command to the RS485 device and the RS485 device can only react according to server commands.



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.

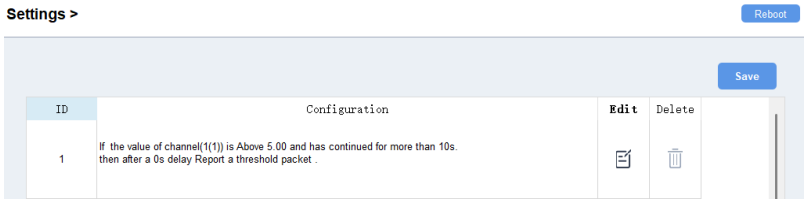


Parameter	Description
Modbus RS485 bridge LoRaWAN®	Enable or disable the Modbus RS485 bridge LoRaWAN® feature.
Pass-through Mode	Select from Active Pass-through or Two-way Pass-through.
Port	The communication port between the RS485 device and the network server. Range: 2-84, 86-223.

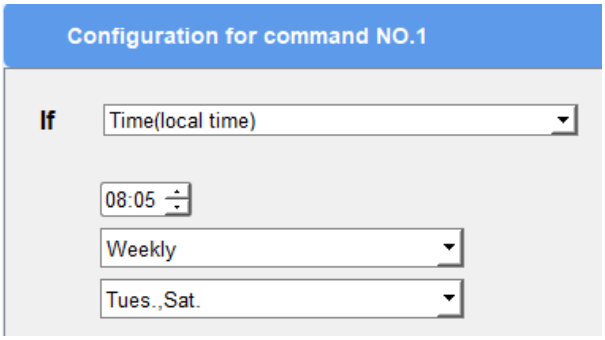
IF-THEN Command

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.



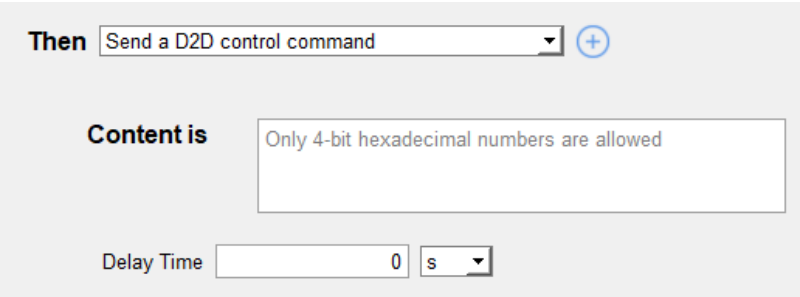
2. Set an IF condition based on the terminal device data or UC100 device status.



Parameter	Description
Time(local time) *(on page 14)*	Set a time.
Channel	When the value of a Modbus channel, reaches the condition. For coil/discrete type, the condition is False/True; For other types, the condition is Above/Within/Below/Change. **Is continued for:** the threshold value should last for some time, the condition is More than/Less than. **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. **Value change time interval:** the change value should last for some time.

Parameter	Description
	Note: The parameter will be hidden if **Two-way Pass-through** feature is enabled.
Received a command via the RS485 interface	Received a command from RS485 devices. **Note:** This parameter is only available when Modbus RS485 bridge LoRaWAN® *(on page 20)* feature is enabled.
Received a server message	Received a custom message from the network server. **Example**: When the custom message is "test", the server should send 74657374 (ASCii to HEX).
Received a Milesight D2D control command	This only works with the Milesight D2D feature is enabled. **Designated state:** After enabled, only the D2D control command with this designed state can trigger UC100. This only works with D2D controller devices which support sending on/off status.
The device restarts	Device is rebooting.

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



Action	Description
Send a server message	Send a custom message to the network server.

Action	Description
Send a Milesight D2D control command	This only works when the Milesight D2D feature is enabled.
Send a command via the RS485 interface	Send a command to the RS485 device.
Upload data packet	Upload the data packet to network server.
Upload alarm packet	Upload a alarm packet to network server.
Restart the Device	Reboot the device.

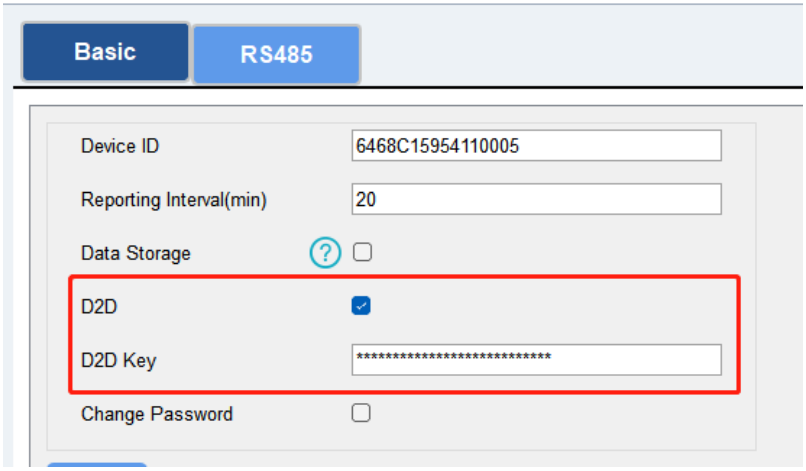
The THEN actions that can be added to the IF condition are as follows:

	Time	Channel	Received a command via the RS485 interface	Received a server message	Received a Milesight D2D control command	The device restarts
Send a server message	√	√	√	√	√	√
Send a Milesight D2D control command	√	√	√	√	√	√
Send a command via the RS485 interface	√	√	√	√	√	√
Upload data packet	√	√		√	√	√
Upload alarm packet		√				
Restart the Device	√	√	√	√	√	

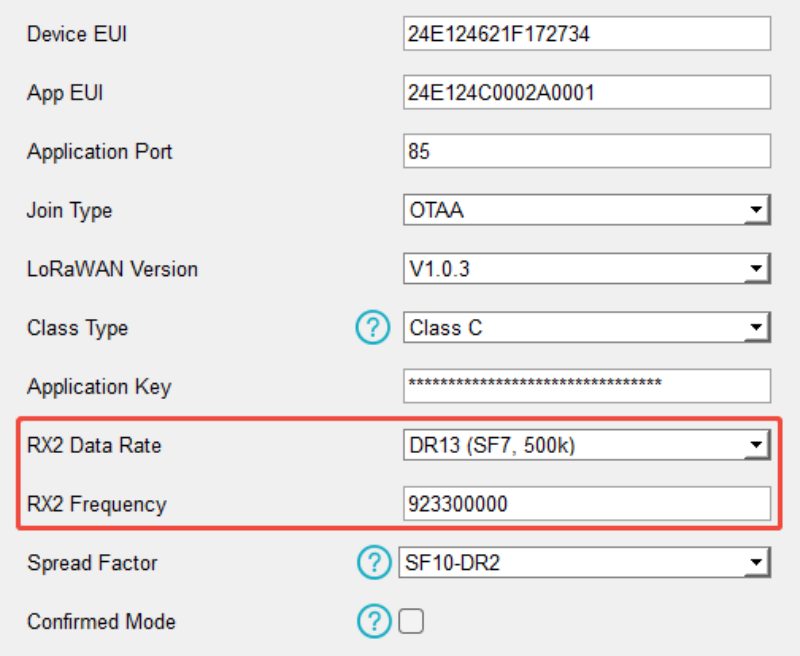
Milesight D2D Settings

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.

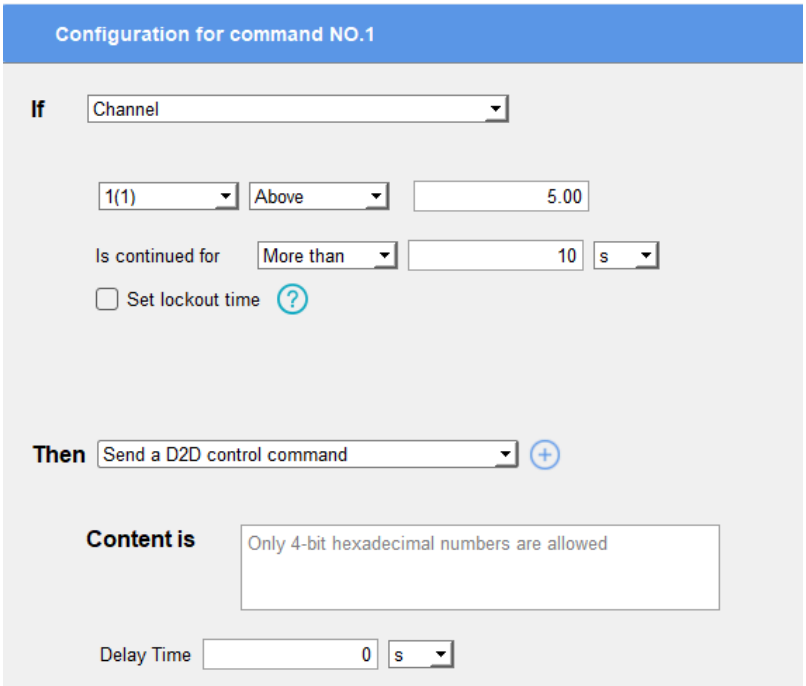
Step 1: Enable Milesight D2D feature and define a unique D2D key that is the same as Milesight D2D controller or agent devices. (Default D2D key: 5572404C696E6B4C6F52613230313823)



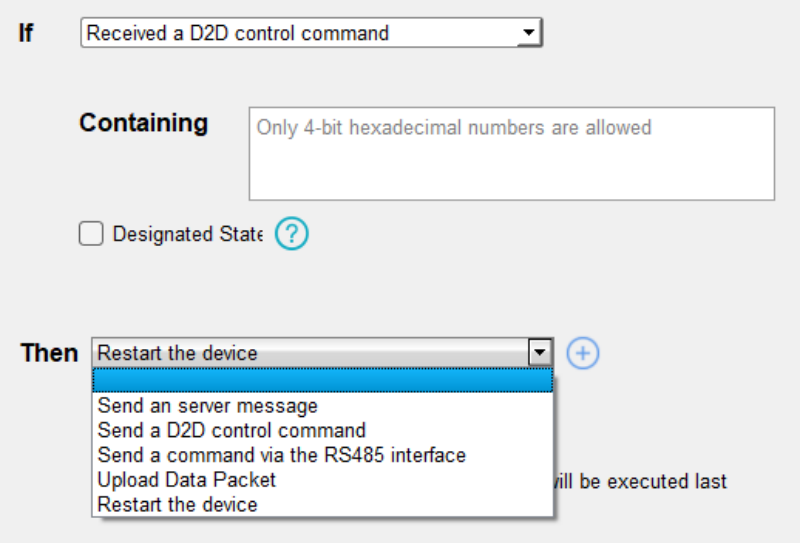
Step 2: Configure the RX2 datarate and RX2 frequency. When UC100 works as Milesight D2D controller, it will send commands as RX2 settings.



Step 3: Command configuration. When the IF condition 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 restart the device, send a server message, send a Milesight D2D control command, send a command via the RS485 interface or upload a data packet.



Maintenance

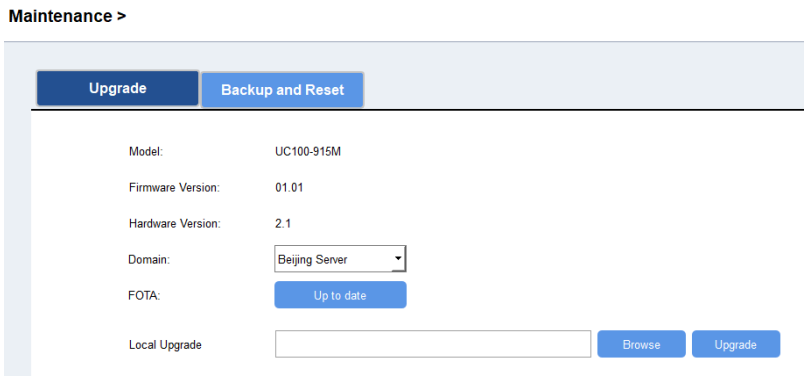
Upgrade

This chapter describes the steps to upgrade the device via ToolBox software.

Step1: Download firmware from Milesight website to your PC.

Step2: 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.

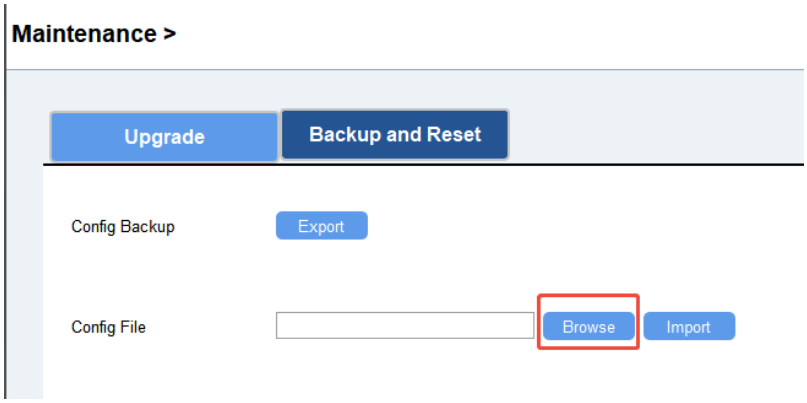


Backup and Restore

This device supports configuration backup for easy and quick device configuration bulk. Backup and restore are allowed only for devices with the same model and frequency band.

Step1: Go to Maintenance**>**Backup and Reset, and click **Export** to save the current configuration as json format backup file.

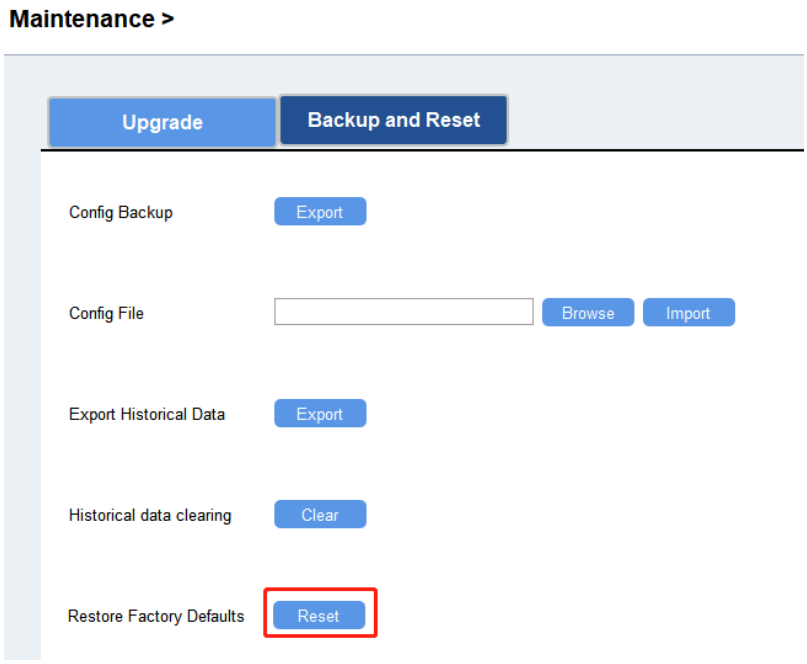
Step2: Click **Browse** to select the backup file, then click **Import** to import the configurations.



Reset to Factory Default

Via Hardware: Hold on the reset button for more than 10s until the LED indicator quickly blinks.

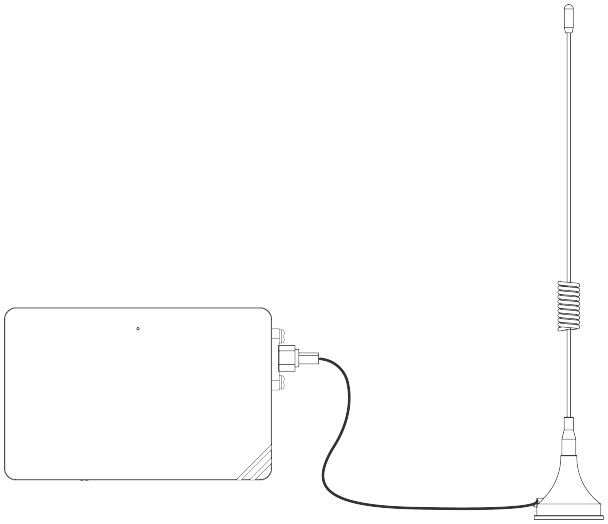
Via ToolBox Software: Go to Maintenance> Backup and Reset to click **Reset**.


Maintenance >
Upgrade
Backup and Reset
Config Backup
Export
Config File
Browse
Import
Export Historical Data
Export
Historical data clearing
Clear
Restore Factory Defaults
Reset

Chapter 5. Installation

Antenna Installation

Rotate the antenna into the antenna connector accordingly.



Note:

1. The antenna should be installed vertically, with the magnetic base attached to a metal surface.
2. Keep the antenna away from walls and ensure there are no obstacles around it.
3. For better coverage, it is recommended to position the antenna at a higher location.
4. Place the antenna near windows when used indoors.
5. Keep a distance of at least 50 cm between antennas.

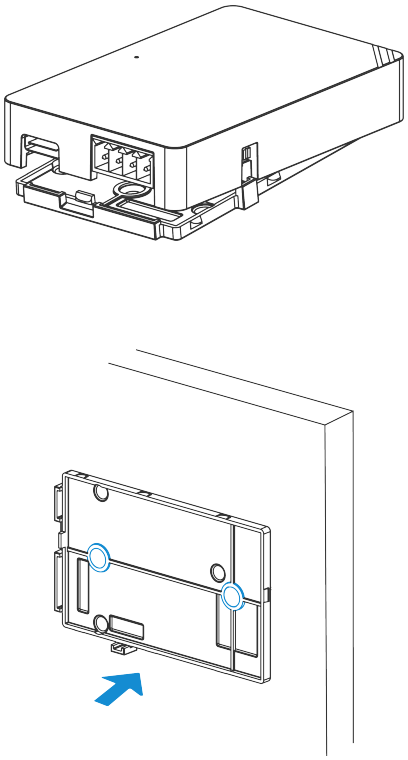
Device Installation

Installation Location

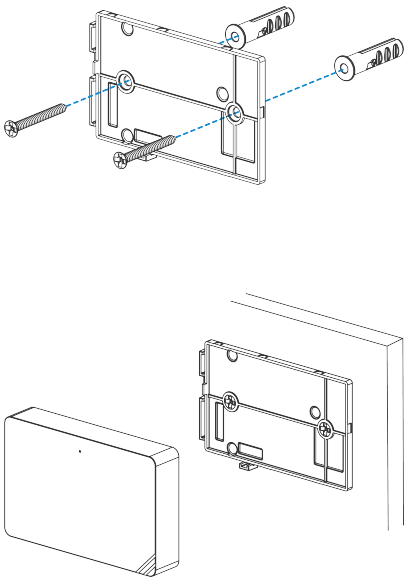
UC100 device can be placed on a desktop or mounted to a wall.

Wall Mounting

Step 1: Remove the device's back cover and securely insert the wall plugs into the designated drilling positions.



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



Chapter 6. Communication Protocol

Overview

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

Channel1	Type1	Data1	Channel2	Type2	Data2	Channel3	...
1 Byte	1 Byte	N Bytes	1 Byte	1 Byte	N Bytes	1 Byte	...

For decoder examples please find files on https://github.com/Milesight-IoT/SensorDecoders.

Uplink Data

This chapter describes the reported data of the device.

Basic Information

UC100 will report a basic information packet whenever joining the network.

Item	Channel	Type	Byte	Description
Power On	ff	0b	1	Device is on
Protocol Version	ff	01	1	Example: 01=V1
Hardware Version	ff	09	2	Example: 03 10 = V3.1
Software Version	ff	0a	2	Example: 03 01 = V3.1
Device Type	ff	0f	1	00: Class A, 01: Class B, 02: Class C
Serial Number	ff	16	8	16 digits
TSL Version	ff	ff	2	Example: 01 00=>V1.0
Reset Event	ff	fe	1	ff, only report when the device resets

Example:

ff0bff ff0101 ffff0201 ff166445b43411300001 ff090200 ff0a0101 ff0f02		
Channel	**Type**	**Value**
ff	0b (Power On)	ff

ff0bff ff0101 ffff0201 ff166445b43411300001 ff090200 ff0a0101 ff0f02		
Channel	**Type**	**Value**
ff	01 (Protocol Version)	01 = V1
ff	ff (TSL version)	0101 = V1.1
ff	16 (Device SN)	6445B43411300001
ff	09 (Hardware Version)	0200 = V2.0
ff	0a (Firmware Version)	0101 = V1.1
ff	0f (Device Type)	02: Class C

Periodic Data

UC100 will report RS485 sensor data which are fetched by Modbus channels according to the 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>f9</td><td>73</td><td>3~10</td><td>Channel ID (1B, 0-31) + Data Type (1B) + Data (Mutable)
Data Type:
Bit 7: 0 = unsigned, 1 = signed
Bit 6-5: 00 = first register, 01 = second register
Bit 4-0:
<table>
<tr><th>Code</th><th>Data Type</th></tr>
<tr><td>00000</td><td>Coil</td></tr>
<tr><td>00001</td><td>Discrete</td></tr>
<tr><td>00010</td><td>Input16</td></tr>
<tr><td>00011</td><td>Hold16</td></tr>
<tr><td>00100</td><td>Hold32</td></tr>
<tr><td>00101</td><td>Hold_float</td></tr>
<tr><td>00110</td><td>Input32</td></tr>
<tr><td>00111</td><td>Input_float</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Item</th><th>Channel</th><th>Type</th><th>Byte</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><th>Code</th><th>Data Type</th></tr>
<tr><td>01000</td><td>Input_int32_with upper 16 bits</td></tr>
<tr><td>01001</td><td>Input_int32_- with lower 16 bits</td></tr>
<tr><td>01010</td><td>Hold_int32_with upper 16 bits</td></tr>
<tr><td>01011</td><td>Hold_int32_with lower 16 bits</td></tr>
<tr><td>01100</td><td>Hold64</td></tr>
<tr><td>01101</td><td>Hold_double</td></tr>
<tr><td>01110</td><td>Input64</td></tr>
<tr><td>01111</td><td>Input_double</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. Range: 0-31</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
...	...
1f	RS485 (Modbus Master) Channel 32

Example:

1. No Modbus channel.

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

2. Fail to fetch Channel1 data.

ff1500		
Channel	**Type**	**Value**
ff	15	00 = Channel 1

3. Succeed to fetch Channel2 first register's data.

f973 0182 ceff				
Channel	**Type**	**Channel ID**	**Data Type**	**Value**
f9	73	01 = Channel 2	82 => 1000 0010 100: Unsigned first register 00010: Input16	ce ff => ff ce = -50

Note:

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

Register Address	Value (Hex)
0	00 15
1	00 20
2	00 25
3	00 30

When using different byte orders and only one register, 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 (0x15002000)	00 20 00 15 (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
Holding/Input Register (INT64)	ABCDEFGH	5911111952302128 (0x0015002000250030)	30 00 25 00 20 00 15 00 (HGFEDCBA)
	GHEFCDAB	13510957797998613 (0x0030002500200015)	15 00 20 00 25 00 30 00 (BADCFEHG)
	BADCFEHG	1513244659789344768 (0x1500200025003000)	00 30 00 25 00 20 00 15 (GHEFCDAB)
	HGFEDCBA	3458805196287644928 (0x3000250020001500)	00 15 00 20 00 25 00 30 (ABCDEFGH)

Alarm Packet

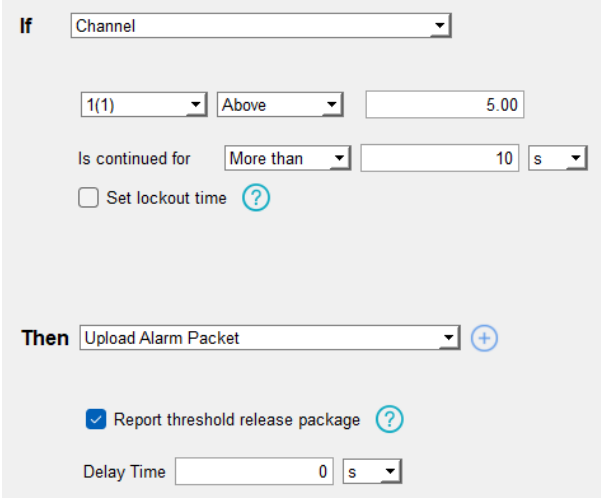
UC100 supports to report Modbus channel threshold alarms, change alarms or custom messages via command settings.

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

Item	Channel	Type	Byte	Description
Modbus Channel Alarm	f9	73	3~10	Alarm Type (1B) + Data Type (1B) + Data (Mutable) **Alarm Type:** Bit 7-6: 01=Threshold alarm, 10=Threshold Alarm release, 11=Change alarm Bit 5-0: Channel ID, range: 0-31 **Data Type:** Bit 7: 0 = unsigned, 1 = signed Bit 6-5: 00 = first register, 01 = second register Bit 4-0: see Data Type *(on page 33)*
Modbus Channel Change Alarm Value	f9	74	9	**Byte 1**: Bit7-6: 00 = first register, 01 = second register Bit5-0: Channel ID, range: 0-31 **Byte 2-9**: Change Value (double)
Custom Message	-	-	1-48	Hex to ASCii

Example:

1. Channel16 data reaches the threshold.



f9 73 4f 85 0040 21c5				
Channel	**Type**	**Channel & Alarm Type**	**Data Type**	**Value**
f9	73	4f => 01001111 01=Threshold alarm 001111: Channel 16	85 => 10000101 100: signed first register 00101: Hold_float	0040 21c5=> c521 4000=-2580

2. Channel32 threshold alarm release.

f973 9f 8f 66666666c687c3c0				
Channel	**Type**	**Channel & Alarm Type**	**Data Type**	**Value**
f9	73	9f => 10011111 10=Threshold Alarm release 011111: Channel 32	8f => 10001111 100: signed first register 01111: Input_double	6666 6666 c687 c3c0 => -9999.55

3. Channel1 first's register change alarm.

f974 43 00000000008040c0 f973c0030f00 f974000000000000002440		
Channel	**Type**	**Value**
f9	73	Alarm Type & Channel: c0=>11 000000, 11=Change alarm, 000000=>0=Channel 1 Data Type: 03=>0 00 00011=Unsigned, first register, Hold16 Value: 0f 00=>00 0f=15
f9	74	Register & Channel: 00 => 00 000000, 00=First register, 000000=>0=Channel 1 Change Value: 00 00 00 00 00 00 24 40=>40 24 00 00 00 00 00 00 =10 (Hex to double)

4. Custom message.

74 65 73 74
Value
74 65 73 74=> t e s t (Hex to ASCii)

Historical Data

The device will report retransmission data or stored data as below format.

<table>
<tr><th>Item</th><th>Chan-nel</th><th>Type</th><th>Byte</th><th>Description</th></tr>
<tr><td>Modbus Channel</td><td>21</td><td>ce</td><td>23</td><td>Byte 1-4: Data unix timestamp, UINT32, Unit: s
Byte 5: Channel ID, Range: 0-31
Byte 6-7: Ctrl
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>3-0</td><td>0000</td></tr><tr><td>5-4</td><td>00=Normal data,
01=Threshold alarm,
10=Threshold alarm release,
11=Change alarm</td></tr><tr><td>7-6</td><td>01: One register
10: Two registers</td></tr><tr><td>8</td><td>0: fetch failure
1: fetch success</td></tr><tr><td>14-9</td><td>Data Type (on page 45)</td></tr><tr><td>15</td><td>0: unsigned, 1: signed</td></tr></table>Byte 8-15: Value 1
Byte 16-23: Value 2</td></tr>
<tr><td>Custom Message</td><td>21</td><td>cd</td><td>4+N</td><td>Byte 1-4: Data unix timestamp, UINT32, Unit: s
Byte 5-N: Message content (Hex to ASCii)</td></tr>
</table>

Example:

1. Historical Modbus channel data.

21ce 0d755b63 01 8085 1400000000000000 ddff000000000000			
Channel	**Type**	**Time Stamp**	**Value**
21	ce	0d 75 5b 63 => 63 5b 75 0d=1666938125s	01: Channel 2 Crtl: 8085 => 1 000010 110000000 Bit15: 1=>signed Bit14-9: 000010 => 02=Input16_AB Bit8: 1=> Fetch success Bit7-6:10 = Two registers Bit5-4:00 => Normal Data Value 1: 14 00 00 00 00 00 00 00=>00 00 00 00 00 00 00 14=>20 Value 2: dd ff 00 00 00 00 00 00=>00 00 00 00 00 00 ff dd => -35

2. Historical custom message.

21cd97e4656874657374			
Channel	**Type**	**Time Stamp**	**Value**
21	cd	97 e4 65 68=>68 65 e4 97=1751508119s	74 65 73 74=> t e s t (Hex to ASCii)

Downlink Command

UC100 supports downlink commands for configuration and control. The downlink application port is 85 by default.

Note:
If the downlink command length is more than 53, please do not send multiple commands together, or change the RX2 Data Rate parameter of the device and network server to a higher value to extend the downlink length limitation.

Basic Settings

<table>
<tr><th>Item</th><th>Chan-nel</th><th>Type</th><th>Byte</th><th>Description</th></tr>
<tr><td>Report Interval</td><td>ff</td><td>03</td><td>2</td><td>UINT16, Unit: s</td></tr>
<tr><td>Reboot</td><td>ff</td><td>10</td><td>1</td><td>ff</td></tr>
<tr><td>Data Storage</td><td>ff</td><td>68</td><td>1</td><td>00: Disable, 01: Enable</td></tr>
<tr><td>Data Retransmission</td><td>ff</td><td>69</td><td>1</td><td>00: Disable, 01: Enable</td></tr>
<tr><td>Data Retrans-mission Interval</td><td>f9</td><td>0d</td><td>2</td><td>UINT16, Unit: s, Range: 30~1200, Default: 600</td></tr>
<tr><td>UTC Time Zone</td><td>ff</td><td>bd</td><td>2</td><td>INT16/60</td></tr>
<tr><td>Sync Time with LNS</td><td>ff</td><td>4a</td><td>1</td><td>00</td></tr>
<tr><td>Daylight Saving Time</td><td>f9</td><td>72</td><td>9</td><td>Byte 1:
<table><tr><th>Bit7</th><th>Bit6-0</th></tr><tr><td>0=Disable, 1=Enable</td><td>DST bias, unit: min, range: 1-120</td></tr></table>Byte 2-5: Start time, Month (1B)+Week&Day (1B) + Minute Time (2B)
Byte 6-9: End time
Week&Day:
<table><tr><th>Bit7-4</th><th>Bit3-0</th></tr><tr><td>Week num-ber, range: 1-5</td><td>Weekday, range: 1-7</td></tr></table></td></tr>
<tr><td>Enquiry Periodic Report</td><td>ff</td><td>28</td><td>1</td><td>ff</td></tr>
<tr><td>Rejoin the Network</td><td>ff</td><td>04</td><td>1</td><td>ff</td></tr>
</table>

Example:

1. Reboot the device.

ff10ff		
Channel	**Type**	**Value**
ff	10	ff

2. Set report interval as 20 minutes.

ff03b004		
Channel	**Type**	**Value**
ff	03	b004=>04b0=1200s=20 minutes

3. Set time zone as UTC-4.

ffbd10ff		
Channel	**Type**	**Value**
ff	bd	10 ff => ff 10 = -240/60=-4

4. Set DST time: start time is last Sunday 1:00 of March, end time is last Sunday 1:00 of October, and bias is 1h (60 minutes).

f972bc03573c000a573c00		
Channel	**Type**	**Value**
f9	72	bc=>1 0111100 =>1=Enable, 0111100=60 minutes Start time: 03=>March, 57=>last (5) Sunday(7), 3c 00 =>00 3c=60 minutes =1:00 End time: 0a=>10=October, 57=>last (5) Sunday(7), 3c 00 =>00 3c=60 minutes =1:00

RS485 Settings

Item	Channel	Type	Byte	Description
Serial Settings	f9	78	7	**Byte 1-4**: Baud rate, options: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 **Byte 5**: Data Bit, options: 07, 08, 09 **Byte 6**: Stop Bit, 01=1, 02=2, 03=1.5

Item	Channel	Type	Byte	Description
				Byte 7: Parity, 00=None, 01=Even, 02=Odd
Modbus Settings	f9	79	7	**Byte 1-2**: Execution interval, Unit: ms, Range: 10-1000 **Byte 3-4**: Max Resp Time, Unit: ms, Range: 10-60000 **Byte 5**: Max Retry Times, Range: 0-5 **Byte 6**: 00=Disable pass-through, 10=Active Pass-through, 11=Two-way Pass-through **Byte 7**: Pass-through port, range: 2-84, 86-223
Enquiry RS485 Settings	f9	7a	1	00=Serial Settings, 01=Modbus Settings

Example:

1. Serial settings: baud rate is 9600, data bit is 8, stop bit is 1, no parity.

f97880250000080100		
Channel	**Type**	**Value**
f9	78	Baud rate: 80 25 00 00=>00 00 25 80=9600 Data bit: 08=8 Stop bit: 01=1 Parity: 00=None

2. Modbus settings: execution interval is 50ms, max resp time is 60000ms, max retry time is 3, enable Active pass-through and set the port as 5.

f979320060ea031005		
Channel	**Type**	**Value**
f9	79	Execution interval: 32 00=>00 32=50ms Max Resp Time: 60 ea => ea 60=60000ms Max Retry Time: 03=3 10=Active Pass-through

f979320060ea031005		
Channel	**Type**	**Value**
		Pass-through port: 05=5

3. Enquire serial settings.

f97a00		
Channel	**Type**	**Value**
f9	7a	00=Enquiry Serial Settings

Reply:

f87a0000 f97880250000080100		
Channel	**Type**	**Value**
f8	7a	00=Serial Settings, 00=Enquire Success
f9	78	Baud rate: 80 25 00 00=>00 00 25 80=9600 Data bit: 08=8, Stop bit: 01=1, Parity: 00=None

Modbus Channel Settings

Item	Channel	Type	Byte	Description
Add/Configure Modbus Channel	ff	ef		01+Channel ID (1B, 1-32)+Slave ID (1B) + Address (2B) + Type (1B) + Quantity & Sign (1B) **Quantity & Sign:** bit4: 1=signed, 0=unsigned (only work with Holding register int or Input register int types) bit3-bit0: Registers number, range: 1-2 **Type:**

Code	Data Type
00	Coil
01	Discrete

Item	Channel	Type	Byte	Description

Code	Data Type
02	Input16_AB
03	Input16_BA
04	Input32_ABCD
05	Input32_BADC
06	Input32_CDAB
07	Input32_DCBA
08	Input32_AB
09	Input32_CD
0a	Input_float_ABCD
0b	Input_float_BADC
0c	Input_float_CDAB
0d	Input_float_DCBA
0e	Hold16_AB
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
1a	Input_double_ABCDEFGH

<table>
<tr><th>Item</th><th>Channel</th><th>Type</th><th>Byte</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><th>Code</th><th>Data Type</th></tr>
<tr><td>1b</td><td>Input_double_GHEFCDAB</td></tr>
<tr><td>1c</td><td>Input_double_BADCFEHG</td></tr>
<tr><td>1d</td><td>Input_double_HGFEDCBA</td></tr>
<tr><td>1e</td><td>Input64_ABCDEFGH</td></tr>
<tr><td>1f</td><td>Input64_GHEFCDAB</td></tr>
<tr><td>20</td><td>Input64_BADCFEHG</td></tr>
<tr><td>21</td><td>Input64_HGFEDCBA</td></tr>
<tr><td>22</td><td>Hold_double_ABCDEFGH</td></tr>
<tr><td>23</td><td>Hold_double_GHEFCDAB</td></tr>
<tr><td>24</td><td>Hold_double_BADCFEHG</td></tr>
<tr><td>25</td><td>Hold_double_HGFEDCBA</td></tr>
<tr><td>26</td><td>Hold64_ABCDEFGH</td></tr>
<tr><td>27</td><td>Hold64_GHEFCDAB</td></tr>
<tr><td>28</td><td>Hold64_BADCFEHG</td></tr>
<tr><td>29</td><td>Hold64_HGFEDCBA</td></tr>
</table>
</td></tr>
<tr><td>Delete Modbus Channel</td><td>ff</td><td>ef</td><td>2</td><td>00+Channel ID (1B, 1-32)</td></tr>
<tr><td>Mobus Channel Name</td><td>ff</td><td>ef</td><td>4-19</td><td>02+Channel ID (1B, 1-32) + Name Length (1B) + Name (Mutable)</td></tr>
</table>

Note:

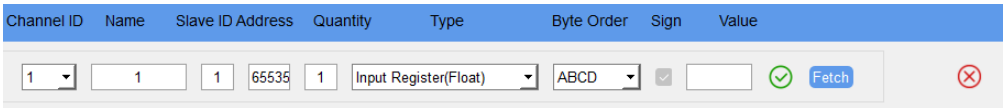
Channel ID in downlink commands is different from uplinks:

Channel ID	Description
01	RS485 (Modbus Master) Channel 1
02	RS485 (Modbus Master) Channel 2

Channel ID	Description
...	...
20	RS485 (Modbus Master) Channel 32

Example:

1. Add a Modbus channel 1: a register, slave ID is 1, address is 1, type is input_float_ABCD.



ffef 01010101000a01		
Channel	**Type**	**Value**
ff	ef	Channel: 01=Channel 1 Slave ID: 01 Address: ff ff =65535 Type: 0a = Input_float_ABCD 01=one register

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

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

Rule Settings

Item	Channel	Type	Byte	Description
Rule Status	f9	76	3	**Byte 1-2**: 1=enable configuration for per bit (rule) **Byte 3**: 01 = Enable, 02 = Disable, 03= Delete
Enquire Rule Settings	f9	77	1	Rule ID, Range: 1~16, the reply content is the same as the configured command

Rule-Condition: *(on page 24)*

<table>
<tr><th>Item</th><th>Chan-nel</th><th>Type</th><th>Byte</th><th>Description</th></tr>
<tr><td>Time</td><td>f9</td><td>7d</td><td>9</td><td>
Byte 1:
<table>
<tr><th>Bit7</th><th>Bit6-0</th></tr>
<tr><td>0=Disable, 1=Enable</td><td>Rule ID, Range: 1-16</td></tr>
</table>
Byte 2: 11
Byte 3: Repeat mode, 00=weekly, 01=monthly
Byte 4-7: Repeat weekday or month day, 1=Enable, 0=Disable for per bit
<table>
<tr><th>Bit</th><th>Weekly Mode</th><th>Monthly Mode</th></tr>
<tr><td>0</td><td>Monday</td><td>1st</td></tr>
<tr><td>1</td><td>Tuesday</td><td>2nd</td></tr>
<tr><td>...</td><td>...</td><td>...</td></tr>
<tr><td>6</td><td>Sunday</td><td>7th</td></tr>
<tr><td>...</td><td>....</td><td>...</td></tr>
<tr><td>30</td><td>0</td><td>30th</td></tr>
</table>
Byte 8: Hour, range: 0-23
Byte 9: Minute, range: 0-59
</td></tr>
<tr><td>Channel</td><td>f9</td><td>7d</td><td>20</td><td>Byte 1:</td></tr>
</table>

<table>
<tr><th>Item</th><th>Chan-nel</th><th>Type</th><th>Byte</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td>
<table>
<tr><th>Bit7</th><th>Bit6-0</th></tr>
<tr><td>0=Disable, 1=Enable</td><td>Rule ID, Range: 1-16</td></tr>
</table>
Byte 2: 12
Byte 3: Channel ID, Range: 1-32
For threshold alarm:
Byte 4:
<table>
<tr><th>Bit7-4: Continue Mode</th><th>Bit3-0: Threshold Mode</th></tr>
<tr><td>0: Is continued for Less than
1: Is continued for More than</td><td>0: False, 1: True
2: Below, 3: Above, 4: Within
Note: False or True is only for Coil or Discrete type.</td></tr>
</table>
Byte 5-8: Continue time, UINT32, Unit: ms, Range: 0-86400000
Byte 9-12: Lock time, UINT32, Unit: ms, Range: 0-86400000
Byte 13-16: Minimum threshold, Float32
Byte 17-20: Maximum threshold, Float32
For change alarm:
Byte 4: 06=Change without time, 07=Change with time interval
Byte 5-12: All 0
Byte 13-16: Value change time interval, UINT32, Unit: ms, Range: 0-86400000
Byte 17-20: Maximum threshold, Float32
</td></tr>
</table>

<table>
<tr><th>Item</th><th>Chan-nel</th><th>Type</th><th>Byte</th><th>Description</th></tr>
<tr><td>Received a com-mand via the RS485 interface</td><td>f9</td><td>7d</td><td>5-51</td><td>Byte 1:<table><tr><th>Bit7</th><th>Bit6-0</th></tr><tr><td>0=Disable, 1=Enable</td><td>Rule ID, Range: 1-16</td></tr></table>Byte 2: 13
Byte 3: Message Length, Range: 2-48
Byte 4-N: Message Content (Hex to ASCii)</td></tr>
<tr><td>Received a server message</td><td>f9</td><td>7d</td><td>5-51</td><td>Byte 1:<table><tr><th>Bit7</th><th>Bit6-0</th></tr><tr><td>0=Disable, 1=Enable</td><td>Rule ID, Range: 1-16</td></tr></table>Byte 2: 14
Byte 3: Message Length, Range: 2-48
Byte 4-N: Message Content (Hex to ASCii), only letter, num-ber, comma, period, separator and exclamation mark are al-lowed</td></tr>
<tr><td>Received a D2D control command</td><td>f9</td><td>7d</td><td>5</td><td>Byte 1:<table><tr><th>Bit7</th><th>Bit6-0</th></tr><tr><td>0=Disable, 1=Enable</td><td>Rule ID, Range: 1-16</td></tr></table>Byte 2: 15
Byte 3-4: D2D Command
Byte 5: Designed state, 00=Disable, 01=On, 02=Off</td></tr>
<tr><td>Device Restart</td><td>f9</td><td>7d</td><td>2</td><td>Byte 1:<table><tr><th>Bit7</th><th>Bit6-0</th></tr><tr><td>0=Disable, 1=Enable</td><td>Rule ID, Range: 1-16</td></tr></table></td></tr>
</table>

Item	Chan-nel	Type	Byte	Description
				Byte 2: 16

Rule-Action: *(on page 24)*

<table>
<tr><th>Item</th><th>Chan-nel</th><th>Type</th><th>Byte</th><th>Description</th></tr>
<tr><td>No Action</td><td>f9</td><td>7d</td><td>2</td><td>Byte 1:
<table><tr><th>Bit7</th><th>Bit6-0</th></tr><tr><td>0=Disable, 1=Enable</td><td>Rule ID, Range: 1-16</td></tr></table>Byte 2: 90=Action 1, a0=Action 2, b0=Action 3</td></tr>
<tr><td>Send a server message</td><td>f9</td><td>7d</td><td>8-55</td><td>Byte 1:
<table><tr><th>Bit7</th><th>Bit6-0</th></tr><tr><td>0=Disable, 1=Enable</td><td>Rule ID, Range: 1-16</td></tr></table>Byte 2: 91=Action 1, a1=Action 2, b1=Action 3
Byte 3-6: Delay time, UINT32, Unit: ms, Range: 0-86400000
Byte 7: Message Length, Range: 1-48
Byte 8-N: Message Content (Hex to ASCii), only letter, number, comma, period, separator and exclamation mark are allowed</td></tr>
<tr><td>Send a D2D control command</td><td>f9</td><td>7d</td><td>8</td><td>Byte 1:
<table><tr><th>Bit7</th><th>Bit6-0</th></tr><tr><td>0=Disable, 1=Enable</td><td>Rule ID, Range: 1-16</td></tr></table>Byte 2: 92=Action 1, a2=Action 2, b2=Action 3
Byte 3-6: Delay time, UINT32, Unit: ms, Range: 0-86400000
Byte 7-8: D2D Command</td></tr>
</table>

<table>
<tr><th>Item</th><th>Chan-nel</th><th>Type</th><th>Byte</th><th>Description</th></tr>
<tr><td>Send a command via the RS485 interface</td><td>f9</td><td>7d</td><td>9-55</td><td>Byte 1:
<table><tr><th>Bit7</th><th>Bit6-0</th></tr><tr><td>0=Disable, 1=Enable</td><td>Rule ID, Range: 1-16</td></tr></table>Byte 2: 93=Action 1, a3=Action 2, b3=Action 3
Byte 3-6: Delay time, UINT32, Unit: ms, Range: 0-86400000
Byte 7: Message Length, Range: 2-48
Byte 8-N: Message Content (Hex to ASCii)</td></tr>
<tr><td>Upload data package</td><td>f9</td><td>7d</td><td>6</td><td>Byte 1:
<table><tr><th>Bit7</th><th>Bit6-0</th></tr><tr><td>0=Disable, 1=Enable</td><td>Rule ID, Range: 1-16</td></tr></table>Byte 2: 94=Action 1, a4=Action 2, b4=Action 3
Byte 3-6: Delay time, UINT32, Unit: ms, Range: 0-86400000</td></tr>
<tr><td>Upload alarm packet</td><td>f9</td><td>7d</td><td>7</td><td>Byte 1:
<table><tr><th>Bit7</th><th>Bit6-0</th></tr><tr><td>0=Disable, 1=Enable</td><td>Rule ID, Range: 1-16</td></tr></table>Byte 2: 95=Action 1, a5=Action 2, b5=Action 3
Byte 3-6: Delay time, UINT32, Unit: ms, Range: 0-86400000
Byte 7: Threshold release packet upload, 00=Dis-able,01=Enable</td></tr>
<tr><td>Device Restart</td><td>f9</td><td>7d</td><td>6</td><td>Byte 1:
<table><tr><th>Bit7</th><th>Bit6-0</th></tr><tr><td>0=Disable, 1=Enable</td><td>Rule ID, Range: 1-16</td></tr></table>Byte 2: 96=Action 1, a6=Action 2, b6=Action 3</td></tr>
</table>

Item	Chan-nel	Type	Byte	Description
				Byte 3-6: Delay time, UINT32, Unit: ms, Range: 0-86400000

Reply Format:

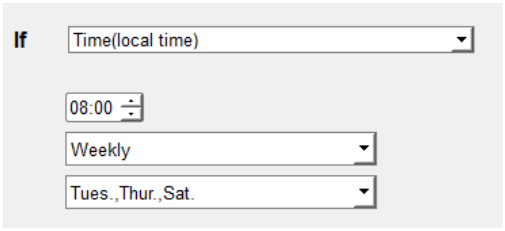
Channel	Type	Command Content	Reply
f8	7d	The same as command	00: Success 02: Out of range 10: Illegal condition 11: Illegal condition parameter 12: Illegal action1 parameter 13: Illegal action2 parameter 14: Illegal action3 parameter 15: Condition conflicts with action1 configuration 16: Condition conflicts with action2 configuration 17: Condition conflicts with action3 configuration 18: Action 1 and Action 2 are configured repeatedly 19: Action 1 and Action 3 are configured repeatedly 1a: Action 2 and Action 3 are configured repeatedly

Example:

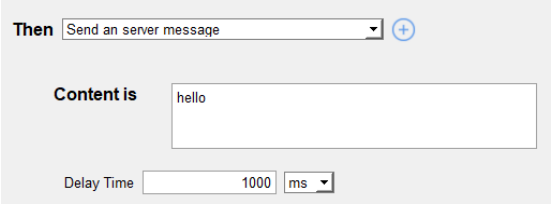
1. Enable Rule 1 and Rule 16.

f976018001		
Channel	**Type**	**Value**
f9	76	01 80 => 80 01=Bit15 and Bit 0 is 1=Rule 16 and Rule 1 01=Enable

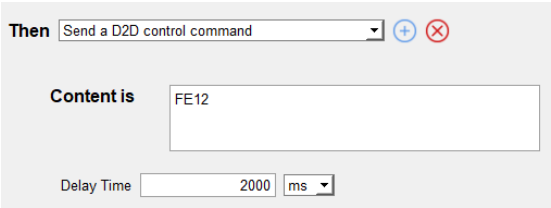
2. Add/Configure Rule 1 as below:



f97d8111002a0000000805		
Channel	**Type**	**Description**
f9	7d	81=>1 0000001=Rule1 enable 11=Time Condition, 00=Weekly 2a000000=>00 00 00 2a = 0010 1010 =every Tues., Thurs., and Sat. 0805=8:05

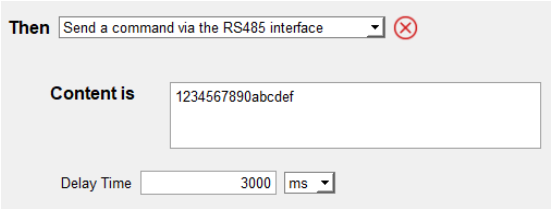


f97d8191e80300000568656c6c6f		
Channel	**Type**	**Description**
f9	7d	81=1 0000001= Rule1 enable 91= Action 1: Send a server message Delay time: e8 03 00 00 => 00 00 03 e8 = 1000ms Message length: 05 => 5 68 65 6c 6c 6f=> hello (Hex to ASCii)



f97d81a2d007000012fe		
Channel	**Type**	**Description**
f9	7d	81=1 0000001= Rule1 enable

f97d81a2d007000012fe		
Channel	**Type**	**Description**
		a2= Action 2: Send a D2D control command Delay time: d0 07 00 00=> 00 00 07 d0 = 2000ms D2D command: 12fe => fe12

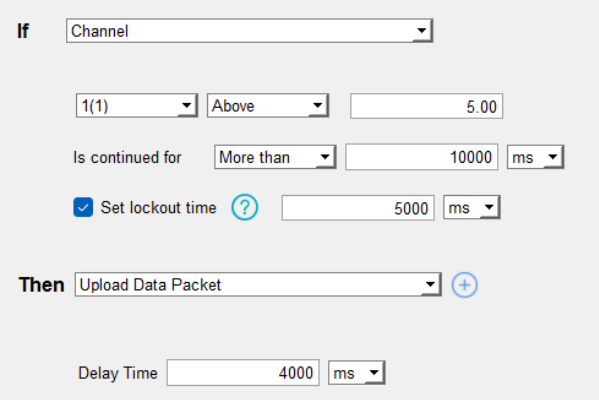


f97d81b3b80b000010313233343536373839306162636465 66		
Channel	**Type**	**Description**
f9	7d	81=1 0000001= Rule1 enable b3= Third Action 3: Send a command via the RS485 interface Delay time: b8 0b 00 00 =>00 00 0b b8=3000ms Message length: 10 =>16 31323334353637383930616263646566 => 1234567890abcdef (Hex to ASCii)

3. Enquire rule 2 configuration.

f97702		
Channel	**Type**	**Value**
f9	77	02=Rule 2

Reply:



f97d8212041310270000881300000000000000a040 f97d8294a00f0000 f97d82a0f97d82b0		
Channel	**Type**	**Value**
f9	7d	82=1 0000010= Rule2 enable 12=Channel Condition, 04=Modbus Channel 4 13=>Above, is more than Continue time: 10 27 00 00 =>00 00 27 10=10000ms Lock time: 88 13 00 00 => 00 00 13 88= 5000ms Minimum threshold: 00 00 00 00

f97d8212041310270000881300000000000000a040 f97d8294a00f0000 f97d82a0f97d82b0		
Channel	**Type**	**Value**
		Maximum threshold: 00 00 a0 40=> 40 a0 00 00=5 (Hex to Float32)
f9	7d	82=1 0000010= Rule2 enable 94= Action 1: Upload a data package Delay time: a0 0f 00 00=>00 00 0f a0=4000ms
f9	7d	82=1 0000010= Rule2 enable a0=Action 2: No Action
f9	7d	82=1 0000010= Rule2 enable b0= Action 3: No action

Historical Data Enquiry

UC100 supports data retrievability feature to send downlink command to enquire the historical data stored in the device. Before that, ensure the device time is correct and data storage feature is enabled to store data.

Command Format:

Item	Channel	Type	Byte	Description
Enquire Data in Time Point	fd	6b	4	Unix timestamp, Unit: s
Enquire Data in Time Range	fd	6c	8	Byte 1-4: Start timestamp, Unit: s Byte 5-8: End timestamp, Unit: s
Stop Query Data Report	fd	6d	1	ff
Data Retrievability Interval	f9	0e	2	UINT16, Unit: s, Range: 30~1200, Default: 60

Reply Format:

Item	Chan-nel	Type	Byte	Description
Enquiry Result	fc	6b/6c	1	00: Enquiry success. The device will report the historical data *(on page 39)* according to data retrievability interval. 01: Time point or time range invalid 02: No data in this time or time range

Note:

1. Use Unix Timestamp Converter to calculate the time.
2. The device only uploads no more than 300 data records per range enquiry.
3. 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's 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:

Enquire the historical data in a time range.

fd6c 64735b63 7c885b63		
Channel	**Type**	**Value**
fd	6c	Start time: 64 73 5b 63 => 63 5b 73 64 = 1666937700s End time: 7c 88 5b 63 => 63 5b 88 7c = 1666943100s

Reply:

fc6c00		
Channel	**Type**	**Value**
fc	6c	00: Enquiry success

21ce 0d755b63 01 8085 1400000000000000 ddff000000000000			
Channel	**Type**	**Time Stamp**	**Value**
21	ce	0d 75 5b 63 => 63 5b 75 0d=1666938125s	01: Channel 2 Crtl: 8085 => 1 000010 110000000 Bit15: 1=>signed Bit14-9: 000010 => 02=Input16_AB Bit8: 1=> Fetch success Bit7-6:10 = Two registers Bit5-4:00 => Normal Data Value 1: 14 00 00 00 00 00 00 00=>00 00 00 00 00 00 00 14=>20 Value 2: dd ff 00 00 00 00 00 00=>00 00 00 00 00 00 ff dd => -35