

# LoRaWAN<sup>®</sup> Controller

## IOT-C50x Series

Communication Protocol

## Revision History

Date	Doc Version	Description
Feb. 2, 2021	V 1.0	Initial version
Dec. 1, 2021	V 1.1	Add collect failed package
Nov. 17, 2022	V 2.0	1. Add RS485 byte order 2. Change analog value decimal place from 2 to 3

## Contents

1. Overview .....	2
2. Uplink Payload .....	3
2.1 Device Information .....	3
2.2 Sensor Data .....	4
3. Downlink Payload .....	7

## 1. Overview

UC50x Series use the standard Milesight IoT payload format based on IPSO. All data are based on following format:

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

Channel	Description
01	Battery
03	GPIO1
04	GPIO2
05	Analog Input1
06	Analog Input2
ff	Device information or RS485 data

### Note:

- 1) All explanations and examples in this document are based on HEX format.
- 2) For all Milesight IoT decoder examples please find files on <https://github.com/Milesight-IoT/SensorDecoders>

## 2. Uplink Payload

Uplink payloads of UC50x Series are made up of device information and sensor data.

### 2.1 Device Information

UC500 series report basic device information of device everytime joining the network.

Channel	Type	Data Size/Byte	Description
ff	01(Protocol Version)	1	01=>V1
	08 (Device SN)	6	12 digits
	09 (Hardware Version)	2	02 10=>V2.1
	0a(Software Version)	2	01 01=>V1.1
	0b (Power On)	1	Device is on
	0f(Device Type)	1	00=Class A 02=Class C
	14 (Analog Type)	1	Bit0~Bit 3: 0: 4-20 mA 1: 0-10 V Bit4~Bit 7: 1: AI1 2: AI2

#### Examples:

ff 0b ff ff 01 01					
Channel	Type	Value	Channel	Type	Value
ff	0b (Power On)	ff (reserved)	ff	01 (Protocol Version)	01(V1)

ff 08 61 22 91 36 34 79		
Channel	Type	Value
ff	08(Device SN)	61 22 91 36 34 79

ff 09 01 20 ff 0a 01 10					
Channel	Type	Value	Channel	Type	Value
ff	09 (Hardware version)	0120 (V1.2)	ff	0a (Software version)	0110 (V1.10)

ff 14 11 ff 14 20							
Channel	Type	Analog Input	Analog Input Signal Type	Channel	Type	Analog Input	Analog Input Signal Type
ff	14 (AI Type)	1 (AI 1)	1 (0-10 V)	ff	14 (AI Type)	2 (AI 2)	0 (4-20 mA)

## 2.2 Sensor Data

UC500 series report sensor data according to reporting interval (10min by default).  
Battery level is reported every 6 hours for UC501 and 12 hours for UC502.

Channel	Type	Data Size/Byte	Description	
01	75(Battery Level)	1	Unit: %	
03(GPIO 1)	00 (Digital Input)	1	00=low, 01=high	
	01 (Digital Output)	1	00=low, 01=high	
	c8(Counter)	4	Unsigned	
04(GPIO 2)	00 (Digital Input)	1	00=low, 01=high	
	01 (Digital Output)	1	00=low, 01=high	
	c8(Counter)	4	Unsigned	
05(AI 1)	02(Analog Input)	8	<b>Byte 1-2:</b> Current value*1000 <b>Byte 3-4:</b> Min value*1000 <b>Byte 5-6:</b> Max value*1000 <b>Byte 7-8:</b> Average value*1000	
06(AI 2)				
ff	0e(RS485)	Mutable (3-6)	Total: Byte 1+Byte 2+Value <b>Byte 1:</b> Channel ID+6 <b>Byte 2:</b> <b>Bit 0~2:</b> Data Type <b>Bit 3~7:</b> Data Length	
			<b>Code</b>	<b>Data Type</b>
			000	Coil
			001	Discrete
			010	Input16 Input_int32_with upper 16 bits Input_int32_with lower 16 bits
011	Hold16 Hold_int32_with upper			

				16 bits Hold_int32_with lower 16 bits
			100	Hold32
			101	Hold_float
			110	Input32
			111	Input_float
ff	15	1	Modbus data collection failed package.The device will return the failed Modbus channel ID.	

**Note:** Channel ID can be configured in ToolBox. If channel ID is 1, device will send 07.

Channel ID	Description
00	RS485 (Modbus Master) Channel 1
01	RS485 (Modbus Master) Channel 2
02	RS485 (Modbus Master) Channel 3
...	
0f	RS485 (Modbus Master) Channel 16

### Examples:

#### 1. Battery

<b>01 75 5a</b>		
Channel	Type	Value
01	75 (Battery)	5a => 90%

#### 2. Digital Input

<b>03 00 00</b>		
Channel	Type	Value
03(GPIO1)	00(Digital Input)	00=>low

#### 3. Digital Output

<b>04 01 01</b>		
Channel	Type	Value
04(GPIO2)	01(Digital Output)	01=>high

#### 4. Pulse Counter

04 c8 78 05 00 00		
Channel	Type	Value
04(GPIO2)	c8(Counter)	78 05 00 00 =>00 00 05 78=1400

#### 5. Analog Input

05 02 c814 c814 c814 c814					
Channel	Type	Ccy Value	Min Value	Max Value	Avg Value
05 (Analog Input1)	02 (Analog Input)	c8 14 => 14 c8 = 5320 means 5.32	c8 14 => 14 c8 = 5320 means 5.32	c8 14 => 14 c8 = 5320 means 5.32	c8 14 => 14 c8 = 5320 means 5.32

#### 6. RS485 regular uplink

ff 0e 07 24 15001500				
Channel	Type	Channel ID	Data Type	Value
ff	0e (RS485)	07 means RS485 (Modbus Master) Channel 1	24 => 0010 0100  Bit0-bit2: 100 means Holding Register (INT32)  Bit3-Bit7: 00100=>4 Means data length = 4	1500 1500 (DCBA)=> 0015 0015 =1376277

**Note:** When data type is holding register or input register, ToolBox can set different byte order. Take following 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)
	BADC	352329728	00 20 00 15

		(0x15002000)	(CDAB)
	CDAB	2097173 (0x00150015)	15 00 20 00 (BADC)
	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)	/	21 (0x15)	15 00 00 00

If UC50x fails to connect Modbus data, it will receive an error message.

<b>ff 15 08</b>		
Channel	Type	Value
ff	15 (collect failed)	08 means RS485(Modbus Master) Channel 2

### 3. Downlink Payload

Downlink xx used for controlling the UC50x via network server remotely. Downlink port (Application port) can be configured via ToolBox.

When the channel range is 1~253, the format is:

Channel1	Data1	Reserved	Channel2	Data2	Reserved	Channel 3	...
1 Byte	2 Byte	ff	1 Byte	2 Byte	ff	1 Byte	...

When the channel is 255(ff), the format is:

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

#### Examples:

- DO Control

<b>04 01 00 ff</b>		
Channel	Data	Reserved
04(GPIO2)	01 00(High)	ff

- Reporting Interval

<b>ff 03 b0 04</b>		
<b>Channel</b>	<b>Type</b>	<b>Value</b>
ff	03(Set Reporting Interval)	b0 04 => 04 b0 = 1200s

**-END-**