



LoRaWAN[®] Controller

UC50x 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
July 7, 2023	V 3.0	Update based on hardware 3.x

Contents

1. Overview	2
2. Uplink Payload	2
2.1 Device Information	2
2.2 Sensor Data	3
2.3 Alarm	7
3. Downlink Payload	9
4. Historical Data Enquiry	10

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	...

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	Byte	Description
ff	01(Protocol Version)	1	01=>V1
	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, 01 = Class B, 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
	16 (Device SN)	8	16 digits

Examples:

ff0bff ff0201 ff1410 ff1421 ff090300 ff0a0105 ff0f02					
Channel	Type	Value	Channel	Type	Value
ff	0b (Power On)	ff (reserved)	ff	01 (Protocol Version)	01 (V1)
Channel	Type	Value	Channel	Type	Value
ff	14 (Analog Type)	10: AI 1=4-20mA 20: AI 2=0-10V	ff	09 (Hardware version)	0300 (V3.0)
Channel	Type	Value	Channel	Type	Value
ff	0a (Software version)	0105 (V1.5)	ff	0f (Device Type)	02 (Class C)

2.2 Sensor Data

UC50x 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.

RS232 uplink doesn't have its own channel or type since it only forwards RS232 terminal devices data to the server transparently.

Channel	Type	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)	e2(Analog Input)	8	Byte 1-2: Current value, float 16 Byte 3-4: Min value, float 16 Byte 5-6: Max value, float 16 Byte 7-8: Average value, float 16																		
06(AI 2)																					
08	db (SDI-12)	37	Byte 1: 00~0f (Channel 1 to 16) Byte 2-37: SDI-12 data (ASCII characters) Note: every channel only reports the first 36 characters and will not report the rest if the data length is more than 36.																		
ff	0e(RS485)	Mutable (3-6)	Total: Byte 1+Byte 2+Value(1~4 Bytes) Byte 1: Channel ID Byte 2: Data Length (bit 7-3)+Data Type (Bit 2-0) Data Type: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Code</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Coil</td> </tr> <tr> <td>001</td> <td>Discrete</td> </tr> <tr> <td>010</td> <td>Input16 Input_int32_with upper 16 bits Input_int32_with lower 16 bits</td> </tr> <tr> <td>011</td> <td>Hold16 Hold_int32_with upper 16 bits Hold_int32_with lower 16 bits</td> </tr> <tr> <td>100</td> <td>Hold32</td> </tr> <tr> <td>101</td> <td>Hold_float</td> </tr> <tr> <td>110</td> <td>Input32</td> </tr> <tr> <td>111</td> <td>Input_float</td> </tr> </tbody> </table>	Code	Data Type	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
Code	Data Type																				
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.

Channel ID	Description
07	RS485 (Modbus Master) Channel 1
08	RS485 (Modbus Master) Channel 2

08	db(SDI-12)	Byte 1: 00=Channel 1 Byte 2-37: Hex to ASCII result is A+0.0+0+25.5
----	------------	--

7. RS485 regular uplink

ff 0e 07 24 15001500				
Channel	Type	Channel ID	Data Type	Value
ff	0e (RS485)	07= RS485 (Modbus Master) Channel 1	24 => 0010 0100 Bit0-bit2: 100=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 the 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 (0x15002000)	00 20 00 15 (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.

Channel ID	Name	Slave ID	Address	Quantity	Type	Byte Order	Sign	Value
1	milesight	1	0	1	Holding Register(INT32)	CDAB	<input type="checkbox"/>	<input type="text"/> <input type="button" value="Fetch"/> <input type="button" value="X"/> <input type="button" value="X"/>

ff 15 08		
Channel	Type	Value
ff	15 (collect failed)	08=RS485(Modbus Master) Channel 2

2.3 Alarm

UC50x series supports adding IF-THEN commands to set and send alarms when the analog value or Modbus channel value reaches the preset conditions. Usually there are two alarm types:

- Threshold alarm: when the value is above or below or within the range of the threshold;
- Change alarm: when the current collected value-last collected value > change value.

Channel	Type	Byte	Description																				
85(AI 1)	e2(Analog Input)	9	Byte 1-2: Current value, float 16 Byte 3-4: Min value, float 16 Byte 5-6: Max value, float 16 Byte 7-8: Average value, float 16 Byte 9: 01=Threshold alarm, 02=Change alarm																				
86(AI 2)																							
80	0e(RS485)	Mutable (4-7)	Total: Channel ID (1 Byte) + Ctrl (1 Byte) + Data(1~4 Bytes)+Alarm Type (1 Byte) Alarm Type: 01=Threshold alarm, 02=Change alarm Ctrl: Data Length (bit 7-3)+Data Type (Bit 2-0) Data Type: <table border="1" data-bbox="833 1243 1332 1944"> <thead> <tr> <th>Code</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Coil</td> </tr> <tr> <td>001</td> <td>Discrete</td> </tr> <tr> <td rowspan="2">010</td> <td>Input16</td> </tr> <tr> <td>Input_int32_with upper 16 bits Input_int32_with lower 16 bits</td> </tr> <tr> <td rowspan="2">011</td> <td>Hold16</td> </tr> <tr> <td>Hold_int32_with upper 16 bits Hold_int32_with lower 16 bits</td> </tr> <tr> <td>100</td> <td>Hold32</td> </tr> <tr> <td>101</td> <td>Hold_float</td> </tr> <tr> <td>110</td> <td>Input32</td> </tr> <tr> <td>111</td> <td>Input_float</td> </tr> </tbody> </table>	Code	Data Type	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
Code	Data Type																						
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																						

Note: Channel ID can be configured in ToolBox.

Channel ID	Description
07	RS485 (Modbus Master) Channel 1
08	RS485 (Modbus Master) Channel 2
09	RS485 (Modbus Master) Channel 3
0a	RS485 (Modbus Master) Channel 4
...	
16	RS485 (Modbus Master) Channel 16

Examples:

1. Analog Input threshold alarm

If

mA

Then

85 e2 9a4a9a4a9a4a9a4a 01						
Channel	Type	Ccy Value	Min Value	Max Value	Avg Value	Alarm Type
85 (Analog Input1)	e2 (Analog Input)	9a 4a => 4a 9a = 13.203mA	9a 4a => 4a 9a = 13.203mA	9a 4a => 4a 9a = 13.203mA	9a 4a => 4a 9a = 13.203mA	01=Thresh old alarm

2. RS485 change alarm

If

Then

80 0e 07130c0002						
Channel	Type	Channel ID	Data Type	Value	Alarm Type	

80	0e (RS485)	07= RS485 (Modbus Master) Channel 1	13 => 0001 0011 Bit0-bit2: 011=Holding Register (INT16) Bit3-Bit7: 00010=>2 Means data length = 2	0c 00=> 00 0c =12	02=Change alarm
----	-------------------	---	---	----------------------	--------------------

3. Downlink Payload

Downlink is used for controlling the UC50x via network server remotely. Downlink port (Application port) is 85 by default and can be configured via ToolBox.

Channel	Type	Description
03 (GPIO1)	/	High level: 01 00 ff
04 (GPIO2)	/	Low level: 00 00 ff
ff	02 (Collecting Interval)	2 Bytes, unit: s
	03 (Reporting Interval)	2 Bytes, unit: s
	10 (Reboot)	ff
	17 (Time Zone)	2 Bytes, UTC timezone * 10
	28 (Get Current Data)	ff
	68 (Data Storage)	00: disable, 01: enable
	69 (Data Retransmission)	00: disable, 01: enable
6a (Data Retransmission Interval)	3 Bytes Byte 1: 00 Byte 2-3: interval time, unit:s range: 30~1200s (600s by default)	

Examples:

1. Change DO2 (GPIO2) status as high.

04 01 00 ff	
Channel	Value
04(GPIO2)	01 00 ff

2. Set the reporting interval as 20 minutes.

ff 03 b0 04		
Channel	Type	Value
ff	03 (Set Reporting Interval)	b0 04 => 04 b0 = 1200 s = 20 mins

3. Reboot the device.

ff10ff		
Channel	Type	Reversed
ff	10 (Reboot)	ff

4. Set time zone as UTC-2.

ff17ecff		
Channel	Type	Value
ff	17	ec ff => ff ec = -20 the time zone is UTC-2

4. Historical Data Enquiry

UC50x 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 Byte 1: 01 Byte 2-3: interval time, unit:s range: 30~1200s (60s by default)

Reply format:

Channel	Type	Description
fc	6b/6c	00: data enquiry success 01: time point or time range invalid 02: no data in this time or time range
20	dc(Basic Historical Data)	Data time stamp (4 Bytes) + GPIO1 Type (1 Byte) + GPIO 1 Data (4 Bytes) + GPIO 2 Type (1 Byte) + GPIO 2 Data (4 Bytes) + AI 1 Current Data (4 Bytes) + AI2 Current Data (4 Bytes)
20	dd (Modbus Channel Historical Data)	Data time stamp (4 Bytes) + Modbus Mask (2 Bytes) + Data (Mutable)
20	e0 (SDI-12 Historical)	Data time stamp (4 Bytes) + SDI-12 Mask (2

	Data)	Bytes) + Data (36 Bytes)
--	-------	--------------------------

Note:

1. GPIO Type: 00=DI, 01=DO, 02=Counter.

Modbus Mask: when bit=1, the corresponding Modbus channel is configured.

Bit	15	...	1	0
Channel	Chan16	...	Chan2	Chan1

Modbus data format: data type (1 Byte) + Value (4 Bytes). The data type format is referred to as follows:

Bit 7-3	Bit 2-0
Data Length	000=Coil, 001=Discrete, 010=Input 16, 011=Hold16, 100=Hold32, 101=Hold_float, 110=Input32, 111=Input_float

SDI-12 Mask: when bit=1, the corresponding SDI-12 channel is configured.

Bit	15	...	1	0
Channel	Chan16	...	Chan2	Chan1

- The device only uploads no more than 300 data records per range enquiry.
- 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 finds 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 2023/07/07 10:15:00 to 2023/07/07 10:25:00.

fd6c 2475a764 7c77a764		
Channel	Type	Value
fd	6c (Enquire data in time range)	Start time: 2475a764 => 64a77524 = 1688696100 =2023/07/07 10:15:00 End time: 7c77a764 => 64a7777c = 1688696700 =2023/07/07 10:25:00

Reply:

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

20 dc 4d75a764 0101000000 0205000000 d0070000 00000000			
Channel	Type	Time Stamp	Value

