 / Doc

https://docs.tuya.com/en/mcu/mcu-protocol.html#

Connection of general serial port of Tuya Cloud

1.Serial communication convention

- Baud rate：9600
- Data bit：8
- Parity check：none
- Stop bit：1
- Data flow control ：none
- MCU：User control panel controls the chip to dock with Tuya Module through serial port

2.Frame Format

Field	Length（byte）	Description
Frame header	2	Fixed as 0x55aa
Version	2	For update extension
Command word	1	Specific frame type
Data length	1	Big end
Data	N	
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Notes：

- All data greater than 1 byte are transmitted in big end mode.
- Generally, the synchronized send-receive mechanism for the same command word is adopted, in which one sends a command, the other will answer. If the sender does not receive the correct response packet on time, it is a transmission timeout.**Note： “Protocol Details” shall prevail for specific communication modes.**

- The asynchronous mode is adopted when sending the module control command and reporting the MCU state. For example, When the “command word” of sending the module control command is x, the “command word”of reporting the MCU state is y.

3.Protocol Details

3.1 Heartbeat Detection

Notes：

- After the module is powered on, the heartbeat is sent every 10s. If no response is received from the MCU within the timeout period (3s), the MCU is considered offline.
- The MCU can also check regularly the module for normal work according to the heartbeat. If the module does not sent any heartbeat, the MCU can reset the module through the hardware reset pin provided by the module.

Module Send:

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x00
Data length	2	0x0000
Data	0	无 None
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

例：0x55aa 00 00 0000 ff

MCU Return

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x00
Data length.	2	0x0001
Data	0	0x00: The return value of the first heartbeat after MCU is restarted is sent only once for the module to check whether the MCU in the work process is restarted; 0x01: Except that 0 is sent for the first time after the MCU is restarted, this value is sent.
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the

Field	Length (byte)	Description
		remainder

Eg.：0x55aa 00 00 0001 00 00（MCU return for the first timed)

```
0x55aa 00 00 0001 01 01(normal return except for the first time)
```

3.2 Query Product Information

Notes：

- prodect key, fixed at 16 bytes, is generated by the Tuya Cloud Developer Platform for the Cloud to record product-related information.
- The product information is composed of prodect key and MCU software versions.

MCU software version number format: In dotted decimal format, “x.x.x” (0<=x<=99), the module also supports “x.x” and ”x” formats and will automatically expand them to “0.x.x” or “0.0.x”.

Module Send:

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x01
Data length	2	0x0000
Data	0	None
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg.：0x55aa 00 01 0000 00

MCU Return：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x01
Data lengt	2	0x0010∽0x0018
Data	0	prodect key+mcu version
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg.: (key=**AIp08kLIftb8x2x0**,mcu ver=**1.0.0**)

0x55aa 00 01 15 **41497030386b4c496674623878327830 312e302e30** 2a

(Note: ASCLL code to hexadecimal, such as “A” to hexadecimal “41”)

3.3 Query MCU and set work modes for the module

Notes: The module work modes mainly cover WIFI work state and the methods to reset WIFI. There are two cases:

1. MCU processing with the module: the MCU reports the reset allocation of network (smartconfig mode or AP mode) to the module, and the module notifies the current work state of WIFI to the MCU through the serial port after receiving the command, and then the MCU performs logic processing to provide different modes for displaying different state supports; the MCU detects the reset demand of WIFI and notifies the module to reset WIFI.
2. Module self-processing: WIFI work state will be displayed in LED through WIFI module’s GPIO pin; WIFI will be reseted according to GPIO input demand of the module. WIFI reset methods under module self-processing: After WIFI module detects the GPIO of the module under lower level at entrance for more than 5s, WIFI will be reseted. The indicator and GPIO pin used by the button will be configured according to the commands below. WIFI reset methods under module self-processing: After WIFI detects the GPIO under lower level at entrance for more than 5s, WIFI will be reseted. The indicator and GPIO pin used by the button will be configured by the commands below.

Module Send:

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x02
Data length	2	0x0000
Data	0	None
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg：0x55aa 00 02 0000 01

MCU Return：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x02
Data length	2	0X0000/0x0002 0x0000: indicates that the module works under MCU processing with the module and MCU needs to perform the functions in the

Field	Length (byte)	Description
		“Note”. 0X0002 indicates that the module works under the module self-processing mode.
Data	0/2	Data length of 2 bytes: the first byte represents GPIO number of WIFI state; the following byte represents GPIO number of WIFI reset key
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg：0x55aa 00 02 0000 01（MCU processing with the module）

```
0x55aa 00 02 0002 0c0d 1a (indicator: GPIO12, reset button: GPIO13)
```

3.4 Report WIFI Work state

Notes：

1. WIFI work state：1、Smartconfig state；2、AP configuration state；3、WIFI configured but Router not connected；4、WIFI configured and Router connected；5、Under the module self-processing mode, corresponding LED will (1) flash 250ms at intervals; (2) flash 1500ms at intervals; (3) stay in long dark state and (4) long bright state
2. When the module detects MCU restart or MCU offline and online again, it will automatically report WIFI status to MCU
3. When the module detects any change of WIFI status, it will automatically report WIFI status to MCU
4. If the module works under the module self-processing mode, MCU does not need to perform the protocol

Module Send：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x03
Data length	2	0x0001
Data	1	Indicate WIFI work state: 0x00: state 1 0x01: state 2 0x02: state 3 0x03: state 4
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg.：0x55aa 00 03 0001 00 03

MCUReturn：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x03
Data length	2	0x0000
Data	0	None
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg.：0x55aa 00 03 0000 02

3.5 Reset WIFI

Notes：

WIFI reset methods under module self-processing: After WIFI detects the GPIO under lower level at entrance for more than 5s, WIFI will be reseted.

If the module works under the module self-processing mode, the MCU does not need to perform the protocol

MCU Send：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x04
Data length	2	0x0000
Data	0	None
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg.：0x55aa 00 04 0000 03

Module Return：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x04
Data length	2	0x0000
Data	0	None

Field	Length (byte)	Description
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg.：0x55aa 00 04 0000 03

3.6 Reset WIFI Selection Mode

Notes：

1. Compared with “3.5 Reset WIFI", this frame enables the MCU to select the configuration mode after WIFI reseting according to its demand
2. MCU access users can selectively perform this protocol
3. If the module works under the module self-processing mode, MCU does not need to perform the protocol

MCU Send：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x05
Data length	2	0x0001
Data	1	0X00: enter smartconifg mode <^/>others: enter AP configuration mode
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

例：0x55aa 00 05 0001 00 05

Module Return：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x05
Data length	2	0x0000
Data	0	None
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg.：0x55aa 00 05 0000 04

3.7 Command Issuing

Notes：

1. Data units for datapoint command/state are as follows:

1) datapoint命令/状态数据单元如下所示：

数据段	长度（byte）	说明			
dpid	1	datapoint序号			
type	1	对应开放平台上某datapoint具体的数据类型,通过如下“表示值”标识			
		类型	表示值	长度（字节）	说明
		raw	0x00	N	对应于raw型datapoint（模块透传）
		bool	0x01	1	value范围：0x00/0x01
		value	0x02	4	对应int类型，大端表示
		string	0x03	N	对应于具体字符串
		enum	0x04	1	枚举类型，范围0-255
		bitmap	0x05	1/2/4	长度大于1字节时，大端表示
len	2	长度对应value的字节数			
value	1/2/4/N	hex表示，大于1字节采用大端传输			

1. For datapoint command/state, data units are “raw” and data units for others are “obj” datapoints
2. “Command issuing” can contain many “Command data units” of datapoint
3. “Command issuing” is an asynchronous treatment protocol corresponding to “state reporting” of datapoint of MCU

Module Send：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x06
Data length	2	It depends on the type and quantity of “Command Data Unit”
Data	N	3.7.1 Command data unit set
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for

Field	Length (byte)	Description
		the remainder

Eg.: The system switch corresponds to No. 3 DP and uses bool type variables and the boot value is 1.

0x55aa 00 06 05 **03 01 0001 01** 10

3.8 state Reporting

Notes：

1.See “3.7” for datapoint state data units.

2.“state reporting” is an asynchronous treatment protocol and there are three trigger mechanisms for “state reporting”:

When the MCU receives the “Command issuing handle frame”, it will execute correctly corresponding datapoint command and then “state reporting”, finally the frame will send the changed datapoint state to the module

After the MCU detects any change of datapoint, it will send the changed datapoint state to the module

When the MCU receives the state query frame in 3.9, all datapoint state will be sent to the module.

3.“Command reporting” can contain many “state data units” of datapoint

MCU Send：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x07
Data length	2	It depends on the type and quantity of “state Data Unit”
Data	N	3.7.1 state data unit set
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg.: the humidity corresponds to No.5 DP, uses valve variables and the humidity is 30℃

0x55aa 00 07 08 **05 02 0004 0000001e** 37

3.9 State Query

Notes：

1. “state query” is an asynchronous treatment protocol used to inquire “obj” type datapoint state of MCU by the module. When the MCU receives the frame, it will report the datapoint state through the reporting frame in 3.8
2. “state query” can be issued as follows:

- After the module is powered on for the first time and connected to the MCU through heartbeat, “state query” can be issued
- In the work process of the module, it detects that the MCU restarts or gets offline and online again, “state query” can be issued

Module Send：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x08
Data length	2	0x0000
Data	0	None
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg.：0x55 aa 00 08 00 00 07

3.10 MCU Update Support

Notes：

1. The MCU can be updated according to its own conditions
2. When to update is triggered by the MCU. The module is only the data transmission channel to support MCU update
3. The update supports the mandatory update and manual update. Specific mode can be selected by uploading update firmware through Tuya Cloud Developer Platform
4. The update only supports the update from lower version to higher version

3.10.1 Update Startup

Notes：

1. There are two update startup modes: automatic and manual mode. Under automatic update mode, when the module detects an updated version of firmware for the cloud MCU, the interaction process for the MCU update packet will be started automatically; under manual update mode and after confirmation through APP, the module will start the interaction process for the MCU update packet.

Module Send：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command	1	0x0a

Field	Length (byte)	Description
word		
Data length	2	0x0004
Data	4	Firmware packet bytes, unsigned int, big end
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg.：0x55aa 00 0a 00 04 00 00 68 00 75(the length of firmware packet 26624, ie. 26KB)

MCU Return：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00/0x01
Command word	1	0x0a
Data length	2	0x0000
Data	0	None
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg.：0x55aa 00 0a 0000 09

Remarks: When the MCU returns to 0x00 version No., it indicates that the transmission displacement of update packet is 2 bytes; When the version No. is 0x01, the transmission displacement of update packet is 4 bytes.

3.10.2 Update packet Transmission

Notes：

1. Update packet transmission data format: packet displacement (unsigned short) + packet data
2. If the MCU receives the frame with its data length of 2 and packet displacement >= firmware size, the packet transmission is completed

Module Send：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x0b
Data length	2	0X0002/0x0004+packet size
Data	N	The front two / four bytes, the firmware is packet displacement
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg.:
(1)If the update is started to return in 0x00 protocol version No., the data packet is

0x55 aa 00 0b 0402 0100 xx…xx XX(packet size 256, packet displacement 0x800)

(2) If the update is started to return in 0x01 protocol version No., the data packet is

0x55aa 00 0b 0404 00000100 xx…xx XX(packet size 256, packet displacement 0x00000800)

MCU Return:

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00/0x01
Command word	1	0x0b
Data length	2	0x0000
Data	0	None
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg.: 0x55aa 00 0b 0000 0a

3.11 Acquire Local Time

MCU Send:

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x1 c
Data length	2	0x0000
Data	0	None
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Eg.: 0x55aa 00 1c 0000 0b

Module Return:

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	0x1 c

Field	Length (byte)	Description
Data length	2	0x0008
Data	7	Data length of 8 bytes: Data [0] indicates whether the time is acquired. 0 means the failed and 1 represents the successful: Data [1] represents the year, 0x00 represents the year of 2000; Data [2] represents the month, from 1 to 12; Data [3] represents the date, from 1 to 31. Data [4] represents the clock, from 0 to 23. Data [5] represents the minute, from 0 to 59; Data [6] represents the second, from 0 to 15; Data [7] represents the week, from 1 to 7 (1 represents Monday)
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

3.12 WIFI Functional Test

Note: Scan specified SSID: Tuya_mdev_test, return to scan results and the Signal strength percentage

MCU Send：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	OxOe
Data length	2	0x0000
Data	Data	None
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Module Return：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	OxOe
Data length	2	0x0002
Data	2	Data length of 2 bytes: Data[0]: 0x00 for failed and 0x01 for successful; When Data[0] is 0x01, Data[1] means the signal strength under the successful mode (0-100, 0 represents the weakest signal and 100 represents the strongest signal); When Data[0] is 0x00, Data [1] is 0x0 which means no specified ssid is found in the scanning under failed mode; When Data[1] is 0x01, it means that the module is not loaded with authorization key
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

3.13 Acquire Module Memory

Note: Acquire remaining memory of WIFI module

MCU Send：

Field	Length (byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	OxOf
Data length	2	0x0000
Data	Data	None
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

Module Return：

Field	Length(byte)	Description
Frame header	2	0x55aa
Version	1	0x00
Command word	1	OxOf
Data length	2	0x0004
Data	4	The data length is 4 bytes with big end format: 0x00 0x00 0x28 0x00 indicates 10240 remaining memory
Checksum	1	Sum by byte from frame header, and the results to be divided by 256 for the remainder

- Index
- Why Tuya
- Product
- APP
- Hardware
- MCU
- Cloud to Cloud
- FAQ

soc_dev
    soc-develop

MCU product develop
    MCU procedure development process

MCU SDK
    mcu-sdkCommunication ProtocolWIFI led & reset

MCU debug

Tuya cloud simulation assistant instructionsTuya MCU simulation debugging assistant instructionsTuya low power cloud assistant simulation instructionsTuya low power MCU simulation debugging assistant instructionsLog query

Production test

Production test

FAQ

MCU develop FAQ

400 Call

Consult

400-881-8611