Zigbee Connection Standard

Hardware Product Development > IoTOS Development > Module SDK

Development Access > Zigbee Protocol Specification

Version: 20210311

Online Version

Contents

Zigbee generic docking is suitable for scenarios where the Zigbee standard protocol is not supported or not very suitable. In this scenario, the Zigbee device mainly acts as a transmission channel between MCU and Zigbee gateway, i.e., it encapsulates the data sent by the gateway in the format of serial protocol and sends it to MCU, and at the same time, it encapsulates the MCU data received from the serial port into the data format of Zigbee ZCL layer and sends it to Zigbee gateway, and then the gateway completes The gateway will then complete the data interaction with the Tuya IoT. In simple terms, the Zigbee network is used to connect the MCU to the Tuya IoT, thus realizing the intelligent upgrade of the product.

1 Terms

Term	Description
Zigbee SDK	Tuya ZigbeeSDK based on CoreTech's Zigbee protocol stack packaged with trimming and optimization.
Attribute	An attribute is a data value that reflects a physical quantity or state
Cluster	Cluster is a cluster that contains one or more attributes (attributes).
EndPoint	EndPoint is the entry point of the application layer of the protocol stack, i.e., the entry address, or the place where the application object exists, which is a set of clusters defined to implement a device description.
Device Id	The serial number defined for each kind of device in Zigbee
Weakly powered device	is a battery-powered device, called a sleep end device in the Zigbee protocol
strong power device	is a device powered by utility power or regulated power from utility power, called router in Zigbee protocol
PID	product ID, each product created in Tuya IoT Platform will generate a unique product number, which is associated with all the information related to the product, such as specific function points, APP control panel, and shipping information.

Term	Description
SoC	system on chip, the hardware itself without MCU, the control program is written into the networking module.
SDK	Software Development Kit, a collection of documents, examples, and tools to assist in the development of a particular type of software. To encourage developers to use their systems or languages, many SDKs are available for free, as is Tuya.
Firmware	Firmware, a program written into an EROM (erasable read-only memory) or EEPROM (electrically erasable programmable read-only memory). Firmware is the device “driver” that is kept inside the device. It is through firmware that the operating system can realize the running action of a specific machine according to the standard device driver, such as optical drives, burners, etc. have internal firmware. Firmware is the software that serves as the most basic and lowest level of work for a system.
OTA	Firmware Over-the-Air, Zigbee module firmware can be upgraded remotely using OTA.
MCU OTA	MCU firmware can be upgraded over the air via Zigbee, which requires the MCU side to support this function.
DP	A set of data formats for data interaction with Tuya IoT, as shown in the table below. ## Zigbee

2 Zigbee device function description

2.1 Zigbee Protocol Version

The product is based on the standard Zigbee 3.0 protocol

Profile Id	0x0104
Device Id	0x0051

2.2 Endpoint

endpoint	description
1	Endpoint for application data interaction

2.3 Supported Clusters

SMART_PLUG (0x0051)

Input Clusters (Sever)

Output Clusters (Client)

Basic (0x0000)

OTA (0x0019)

Time (0x000A)

Private cluster (0XEF00)

2.3.1 Basic Cluster

Attributes:

ID	name	Data Type	Range	Default
0x0000	ZCLVersion	uint8 -0x20	0x00-0xff	0x03

ID	name	Data Type	Range	Default	
0x0001	Application Version	uint8 -0x20	0x00-0xff	ie: 0b 01 00 0001 = 1.0.1 ie 0x41 for 1.0.1 OTA function will use this version number, at the beginning of the OTA phase, the gateway will read the The version number of the OTA package will be read by the gateway and pushed to the device at the beginning of OTA. After a success-ful OTA reboot, the gateway will read	0x0002

ID	name	Data Type	Range	Default		
0x0002	StackVersic	uint8 -0x20	0x00-0xff	0x02		
0x0003	HWVersion	uint8 -0x20	0x00-0xff	0x01		
0x0004	Manufactur	string -0x42	0-32 bytes	XXX...XXX (16 bytes in length, consist- ing of an 8-byte prefix and an 8-byte PID) 0-7 bytes: * TZE600* 8-16 bytes: PID (created and provided by the product manager in the platform or self- service)	0x0005	Modle Id

ID	name	Data Type	Range	Default
0x0005	Modle Identifier	string -0x42	0-32 bytes	TS0105 This field is used for the gateway to quickly identify the device type and enhance the expe- rience
0x0007	Power Source	enum8- 0x30	0x00-0xff	depends on your product
0xfffd	Cluster Revision	uint16 -0x21	0x0000- 0xffff	0x0001

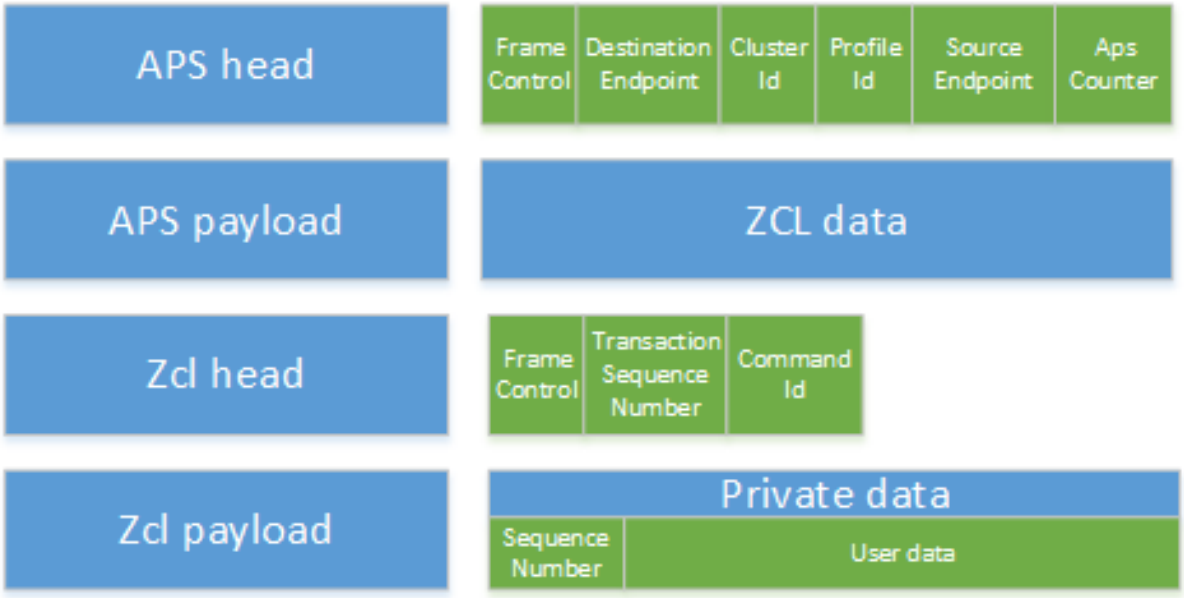
Commands:

ID	name	Direction	Description
0x00	Reset to factory default	C->S	C : client S : server

2.3.2 Private cluster

In order to meet the requirements of generic interfacing, a private cluster is used and a private cluster command is defined for data transfer to achieve data interaction. The Destination Endpoint : 0x01, ClusterID : 0xEF00, Profile ID : 0x0104, Source Endpoint : 0x01 in the APS layer.

In the zcl layer, the private command id is used to represent the different data interaction commands, and the private frame format is defined in the ZCL payload in the form of sequence number (2byte) + DP data, and the data format of DP data is shown in the table of DP data formats.



The frame format of the Zcl head can be found in the following table.

field	length(bit)	value	description
Frame type	2	01	Command is Specific to a Cluster
Manufacture specific	1	0	Manu Code Not included in zcl
Direction	1	0/1	0: GW->Zigbee 1: Zigbee->GW

field	length(bit)	value	description
Disable default response	1	0/1	The default is 1, which is 0 only when the device data is actively reported, then the gateway will actively return a Response message as a marker of successful reporting.
Reserved	3	000	Reserved

2.3.3 Private command id

Command Enumeration	Value	Description
TY*DATA* REQUEST	0x00	Gateway-side data request
TY_DATA_RESPONSE	0x01	Reply to MCU-side data request
TY_DATA_R	0x02	MCU-side data active upload (bidirectional)

Command Enumeration	Value	Description
TY_DATA_QUERY	0x03	GW send, trigger MCU side to report all current information, no zcl payload. Note: Device side can make a policy, data better not to report centrally
TUYA_MCU_	0x10	Gw->Zigbee gateway query MCU version
TUYA_MCU_VERSION_RSP	0x11	Zigbee->Gw MCU return version or actively report version

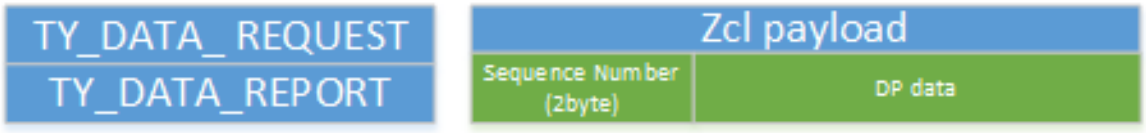
Command Enumeration	Value	Description
TUYA_MCU_	0x12	Gw->Zigbee gateway notifies MCU of upgrade
TUYA_OTA_BLOCK_DATA_REQ	0x13	Zigbee->Gw requests an upgrade package for the MCU
TUYA_OTA_	0x14	Gw->Zigbee gateway returns the requested upgrade package
TUYA_MCU_OTA_RESULT	0x15	Zigbee->Gw returns the upgrade result for the mcu

| TUYA_OTA_BLOCK_DATA_REQ | 0x13 | Zigbee->Gw requests an upgrade package for the MCU |

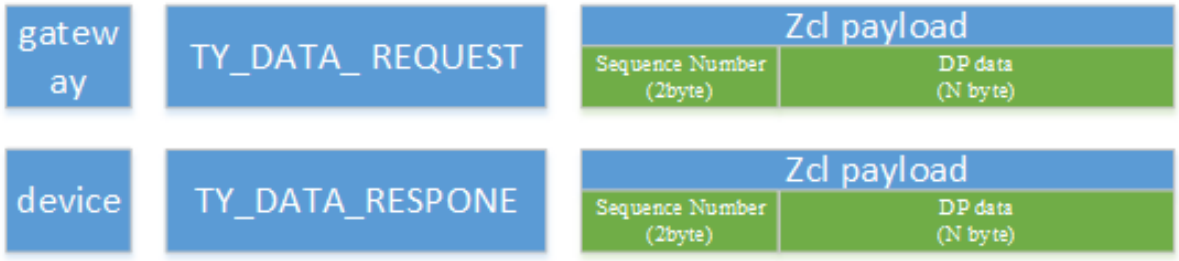
Command Enumeration	Value	Description
TUYA_MCU_	0x24	Time synchronization (bidirectional)

2.3.4 Frame of command id

- When the device receives the data from the gateway, it sends it to the MCU through the corresponding serial protocol command, and then the MCU will process it and reply to the module through the serial protocol after receiving it, and then the module will convert it to TY_DATA_ RESPONSE after receiving it, completing a DP transmission. RESPONSE command to the gateway to complete a DP transmission.



- When the local state of the MCU side changes and needs to be reported to the gateway, the module uses the TY_DATA_REPORT command to report, and the gateway uses the same command to reply after receiving it.



- MCU version number query

TUYA_MCU_VERSION_REQ	Zcl payload: Sequence Number (2byte)
TUYA_MCU_VERSION_RSP	Zcl payload: Sequence Number (2byte) | Version (1byte)

- MCU OTA upgrade

TUYA_MCU_OTA_NOTIFY	Zcl payload: Sequence Number (2byte) | Key (8byte) | version (1byte) | Image size (4byte) | CRC (4byte)
TUYA_OTA_BLOCK_DATA_REQ	Zcl payload: Sequence Number (2byte) | Key (8byte) | version (1byte) | offset (4byte) | size (4byte)
TUYA_OTA_BLOCK_DATA_RSP	Zcl payload: Sequence Number (2byte) | status (1byte) | Key (8byte) | version (1byte) | offset (4byte) | Image data (≤50byte)
TUYA_MCU_OTA_RESULT	Zcl payload: Sequence Number (2byte) | status (1byte) | Key (8byte) | version (1byte)

- Time synchronization

device	TUYA_MCU_SYNC_TIME	Zcl payload: Sequence Number (2byte) | Standard timestmp (4byte)(0) | local timestmp (4byte)(0)
gateway	TUYA_MCU_SYNC_TIME	Zcl payload: Sequence Number (2byte) | Standard timestmp (4byte) | local timestmp (4byte)

DP Data Format

Data segment

Length (byte)

Description

DPID

1

Datapoint Serial Number

type

Corresponds to the specific data type of a datapoint on the open platform, identified by the following “denoted value”

```
<tr>
    <td>Type</td>
    <td>Represented Value</td>
    <td>Length in bytes</td>
    <td>Description</td>
</tr>
<tr>
     <td>raw</td>
    <td>0x00</td>
    <td>N</td>
    <td>Corresponds to raw datapoint (module pass-through)</td>
</tr>
<tr>
     <td>bool</td>
    <td>0x01</td>
    <td>1</td>
    <td> value range: 0x00/0x01</td>
</tr>
<tr>
     <td>value</td>
    <td>0x02</td>
    <td>4</td>
    <td>corresponds to int type, big end representation</td>
</tr>
<tr>
     <td>string</td>
    <td>0x03</td>
    <td>N</td>
    <td>corresponds to a specific string</td>
</tr>
<tr>
     <td>enum</td>
    <td>0x04</td>
    <td>1</td>
    <td>Enumeration type, range 0-255</td>
</tr>
<tr>
     <td>bitmap</td>
    <td>0x05</td>
    <td>1/2/4</td>
    <td>Large end representation for lengths greater than 1 byte</td>
</tr>
<tr>
    <td >len</td>
    <td >2</td>
    <td colspan="3">Length corresponds to the number of bytes of value
       </td>
</tr>
<tr>
    <td >Value</td>
     <td >1/2/4/N</td>
    <td colspan="4">hex indicates that transfers greater than 1 byte
       are big-endian</td>
</tr>
```

2.3.5 Over The Air Upgrade

Attributes:

ID	name	Data Type	Range	Default
0x0000	Upgrade ServerID	EUI64-0xF0	—	0xffffffffffffffffffff
0x0001	File Offset	uint32-0x23	—	0x00000000
0x0002	Current File Version	uint32-0x23	—	0x21050002
0x0006	Image Upgrade Status	enum8-0x30	—	0x00
0x0007	Manufacturer ID	uint16 -0x21	—	0x1168
0x0008	Image Type ID	uint16 -0x21	—	0x80f6
0x0009	Min Block Request Period	uint16 -0x21	—	0x0000
0xfffd	Cluster Revision	int16-0x29	0x0000-0xffff	0x0001

Commands:

ID	name	Direction
0x00	Image Notify	S->C
0x01	Query Next Image Request	C->S
0x01	Query Next Image Request	C->S
0x03	Image Block Request	C->S

ID	name	Direction
0x03	Image Block Request	C->S
0x03	Image Block Request	C->S
0x06	Upgrade End Request	C->S

2.4 Serial Protocol

The data frame for UART communication between the Tuya Zigbee module and the MCU consists of a Front, a Ver, a Cmd, a Length, a Data, and a Check, defined and described as follows

Octets: 2	1	2	1	2	Variable	1
Front	Ver	Seq	Cmd	Length	Data	Check

Frame Format Description

Field	Description
Frame Header (Front)	2 bytes of leading characters, fixed to 0x55aa
Version (Ver)	Serial communication protocol version, 0x02 for upgrade extension
Sequence number (seq)	Sequence number of transmitted data, range 0-65535, revert to 0 after reaching 65535
Command word (Cmd)	The specific frame type, refer to the following table
Length	The valid length of the data to be transmitted; the length of a single frame should not exceed 64 bytes.
Data	The valid data to be transmitted

Field	Description
Check	Check the data bytes from the header of the frame and the result will be rounded to 256.

Serial command list

Cmd ID	Description
0x01	Product information query/reporting
0x02	Device status query/reporting
0x03	Zigbee Device Reset
0x04	Command Issuance
0x05	Status reporting
0x06	Status query
0x07	reserved
0x08	Zigbee device function test
0x09	Query key information (only available for scene switch type devices)
0x0A	Scene wakeup command (only available for Scene switch type devices)
0x0A-0x23	reserved
0x24	time synchronization

3 Zigbee device flow mechanism

3.1 Requirements for the networking process

Normally powered devices

- MCU sends configuration command to turn on the module mapping
- Beacon request channel scan interval is greater than 200ms, priority scanning priority channel
- Scan time is less than 3 minutes
- If data reporting is required, a random delay of 5s to 10s or more is used to report data after a successful networking

Low-power devices

- MCU sends configuration command to enable module networking
- Beacon request channel scan interval is greater than 200ms, priority scanning priority channel
- Scan time is less than 3 minutes
- After successful mapping, data request time is 250ms for 1 minute, then switch to 5s.

3.2 Requirement for device power-off start-up process

At least 15s after the data is reported

3.3 Requirements for link maintenance

Normally powered devices

- Random value between 1 hour and 2 hours to report device version number
- Random value reporting device version number between 1 hour and 2 hours delayed after any data is successfully reported

Low power device

The module needs to report the heartbeat in about 2 minutes, with the device version number in the heart state

3.4 Requirements for data upload

- If APS layer retransmission is enabled it is recommended to set Disable Default Response to 1 to avoid too much ack.
- The Transaction Sequence Number of retransmitted data should be consistent.
- The retransmission interval is greater than 250ms, and the retransmission duration is recommended to be less than 3s.
- When you receive broadcast, multicast, or situational control, you need to delay 1 minute - 2 minutes to report the status of the random value, and report the status immediately when you receive unicast control during the delay time is not reached

4 Development method

4.1 Development based on Tuya Zigbee SDK

- You only need to modify the module type and hardware configuration of the demo to complete the above functions, including groups, scenarios are realized
- Tudou provides complete burn-in, licensing, production testing process, and hardware and software tools
  - Tuya Zigbee SDK Instructions for Use
  - Burn-in license, production test tool instructions scope/production-test-tool/introduction-of-tools)

4.2 Self-developed - non-Tuya Zigbee module

- Complete implementation of the above technical details is required
- Complete device authorization
  - Licensed using the Tuya full licensing tool
  - Authorization via serial communication between the module and the host computer.
  - Module authorization using Tuya Authorization Uplink; scope/production-test-sdk/zigbee-devices-sdk)

4.3 Self-developed-Tuya Zigbee module

- Only the firmware functionality details need to be implemented
- Just purchase the licensed Tuya Zigbee module