**Tuya IPC SDK Description**

|  |  |
| --- | --- |
| Update Time | SDK Version |
| 2019-04-26 | 4.3.2 |
|  |  |
|  |  |
|  |  |
|  |  |

Contents

[1 Macro Definitions 6](#_Toc8848460)

[1.1.1 Macro Definitions for Device Information 6](#_Toc8848461)

[1.1.2 Error Code Macro Definitions 6](#_Toc8848462)

[1.1.3 Constant Macro Definitions 10](#_Toc8848463)

[1.1.4 Macro Definitions of the Group and Scenario Error Code 10](#_Toc8848464)

[1.1.5 DP Message Macro Definitions 11](#_Toc8848465)

[1.1.6 Other Macro Definitions 15](#_Toc8848466)

[2 Enum Definitions 16](#_Toc8848467)

[2.1.1 Types of Uploaded Notification Content 16](#_Toc8848468)

[2.1.2 IPC Registration States 16](#_Toc8848469)

[2.1.3 Gateway States 17](#_Toc8848470)

[2.1.4 Input Types of Direct Network Access 17](#_Toc8848471)

[2.2.1 Video Code Rate 17](#_Toc8848472)

[2.2.2 Audio and Video Coding Protocols 18](#_Toc8848473)

[2.2.3 Audio Sampling Rate 19](#_Toc8848474)

[2.2.4 Audio Bits 20](#_Toc8848475)

[2.2.5 Audio Channels 20](#_Toc8848476)

[2.2.6 Cloud Storage Order Types 20](#_Toc8848477)

[2.2.7 Event Types 21](#_Toc8848478)

[2.2.8 Event States 21](#_Toc8848479)

[2.3.1 Writing Files to an SD Card 21](#_Toc8848480)

[2.3.2 Deleting Files 22](#_Toc8848481)

[2.3.3 Storage States 22](#_Toc8848482)

[2.3.4 Event Playback 23](#_Toc8848483)

[2.3.5 Frame Types 23](#_Toc8848484)

[2.3.6 Playback States 24](#_Toc8848485)

[2.4.1 Reset Types 24](#_Toc8848486)

[2.5.1 Maximum Number of Video and Audio Stream Channels 24](#_Toc8848487)

[2.5.2 User Index 25](#_Toc8848488)

[3 Structure Definitions 26](#_Toc8848489)

[3.1.1 IPC Environment Variables 26](#_Toc8848490)

[3.1.2 TY\_INIT\_PARAMS\_S 27](#_Toc8848491)

[3.1.3 GW\_ATTACH\_ATTR\_T 27](#_Toc8848492)

[3.1.4 DEV\_DESC\_IF\_S 27](#_Toc8848493)

[3.1.5 DP\_PROP\_VAL\_S 28](#_Toc8848494)

[3.1.6 DP\_PROP\_ENUM\_S 28](#_Toc8848495)

[3.1.7 DP\_PROP\_STR\_S 29](#_Toc8848496)

[3.1.8 DP\_BOOL\_S 29](#_Toc8848497)

[3.1.9 DP\_PROP\_BITMAP 29](#_Toc8848498)

[3.1.10 TY\_RECV\_OBJ\_DP\_S 30](#_Toc8848499)

[3.1.11 TY\_OBJ\_DP\_S 30](#_Toc8848500)

[3.1.12 TY\_DP\_QUERY\_S 31](#_Toc8848501)

[3.1.13 FW\_UG\_S 31](#_Toc8848502)

[3.2.1 IPC\_MEDIA\_INFO\_S 31](#_Toc8848503)

[3.2.2 AES\_HW\_CBC\_FUNC 32](#_Toc8848504)

[3.3.1 SS\_QUERY\_DAY\_TS\_ARR\_S 33](#_Toc8848505)

[3.3.2 SS\_FILE\_TIME\_TS\_S 33](#_Toc8848506)

[3.4.1 Ring\_Buffer\_Node\_S 34](#_Toc8848507)

[3.4.2 S\_PRESET\_PTZ 35](#_Toc8848508)

[3.4.3 S\_PRESET\_POSITION 35](#_Toc8848509)

[3.5 Joint Type Definition 36](#_Toc8848510)

[3.5.1 TY\_OBJ\_DP\_VALUE\_U 36](#_Toc8848511)

[4 Interface Definitions 36](#_Toc8848512)

[4.1 General Interface Definitions 36](#_Toc8848513)

[4.1.1 tuya\_ipc\_get\_sdk\_info 36](#_Toc8848514)

[4.1.2 tuya\_ipc\_init\_sdk 37](#_Toc8848515)

[4.1.3 tuya\_ipc\_start\_sdk 38](#_Toc8848516)

[4.1.4 tuya\_ipc\_upgrade\_sdk 38](#_Toc8848517)

[4.1.5 tuya\_ipc\_get\_register\_status 40](#_Toc8848518)

[4.1.6 DEV\_OBJ\_DP\_CMD\_CB 40](#_Toc8848519)

[4.1.7 DEV\_DP\_QUERY\_CB 41](#_Toc8848520)

[4.1.8 GW\_STATUS\_CHANGED\_CB 42](#_Toc8848521)

[4.1.9 GW\_UG\_INFORM\_CB 42](#_Toc8848522)

[4.1.10 TUYA\_RST\_INFORM\_CB 43](#_Toc8848523)

[4.1.11 TUYA\_RESTART\_INFORM\_CB 44](#_Toc8848524)

[4.1.12 GET\_FILE\_DATA\_CB 44](#_Toc8848525)

[4.1.13 UPGRADE\_NOTIFY\_CB 45](#_Toc8848526)

[4.2 Time Interface Definitions 46](#_Toc8848527)

[4.2.1 tuya\_ipc\_get\_service\_time 46](#_Toc8848528)

[4.2.2 tuya\_ipc\_check\_in\_dls 47](#_Toc8848529)

[4.2.3 tuya\_ipc\_get\_local\_time 48](#_Toc8848530)

[4.2.4 tuya\_ipc\_get\_utc\_tm 49](#_Toc8848531)

[4.2.5 tuya\_ipc\_get\_utc\_time 50](#_Toc8848532)

[4.3 Message Interface Definitions 51](#_Toc8848533)

[4.3.1 tuya\_ipc\_dp\_report 51](#_Toc8848534)

[4.3.2 tuya\_ipc\_dp\_report\_sync 52](#_Toc8848535)

[4.3.3 tuya\_ipc\_notification\_content\_upload\_from\_buffer 53](#_Toc8848536)

[4.3.4 tuya\_ipc\_notification\_content\_upload\_from\_file 54](#_Toc8848537)

[4.3.5 tuya\_ipc\_notification\_message\_upload 55](#_Toc8848538)

[4.3.6 tuya\_ipc\_snapshot\_message\_upload 56](#_Toc8848539)

[4.3.7 tuya\_ipc\_report\_living\_msg 56](#_Toc8848540)

[4.4 Device Management Interface Definitions 57](#_Toc8848541)

[4.4.1 tuya\_ipc\_get\_wakeup\_data 57](#_Toc8848542)

[4.4.2 tuya\_ipc\_get\_heartbeat\_data 58](#_Toc8848543)

[4.4.3 tuya\_ipc\_direct\_connect 59](#_Toc8848544)

[4.4.4 tuya\_ipc\_set\_net\_info\_nofify\_cb 60](#_Toc8848545)

[4.4.5 tuya\_ipc\_get\_mqtt\_status 61](#_Toc8848546)

[4.4.6 tuya\_ipc\_book\_wakeup\_topic 62](#_Toc8848547)

[4.4.7 tuya\_ipc\_get\_mqtt\_socket\_fd 63](#_Toc8848548)

[4.4.8 tuya\_ipc\_set\_log\_attr 63](#_Toc8848549)

[4.4.9 tuya\_ipc\_get\_free\_ram 65](#_Toc8848550)

[4.5 Cloud Storage Interface Definitions 65](#_Toc8848551)

[4.5.1 tuya\_ipc\_cloud\_storage\_init 65](#_Toc8848552)

[4.5.2 tuya\_ipc\_cloud\_storage\_uninit 66](#_Toc8848553)

[4.5.3 tuya\_ipc\_cloud\_storage\_get\_store\_mode 67](#_Toc8848554)

[4.5.4 tuya\_ipc\_cloud\_storage\_event\_start 68](#_Toc8848555)

[4.5.5 tuya\_ipc\_cloud\_storage\_event\_stop 69](#_Toc8848556)

[4.5.6 tuya\_ipc\_cloud\_storage\_get\_event\_status 70](#_Toc8848557)

[4.5.7 tuya\_ipc\_cloud\_storage\_pause 70](#_Toc8848558)

[4.5.8 tuya\_ipc\_cloud\_storage\_resume 71](#_Toc8848559)

[4.5.9 tuya\_ipc\_cloud\_storage\_event\_add 72](#_Toc8848560)

[4.5.10 tuya\_ipc\_cloud\_storage\_event\_delete 73](#_Toc8848561)

[4.6 SD Card Storage Interface Definitions 74](#_Toc8848562)

[4.6.1 tuya\_ipc\_ss\_init 74](#_Toc8848563)

[4.6.2 tuya\_ipc\_ss\_uninit 75](#_Toc8848564)

[4.6.3 tuya\_ipc\_ss\_set\_write\_mode 75](#_Toc8848565)

[4.6.4 tuya\_ipc\_ss\_get\_write\_mode 76](#_Toc8848566)

[4.6.5 tuya\_ipc\_ss\_trigger\_event 77](#_Toc8848567)

[4.6.6 tuya\_ipc\_ss\_start\_event 78](#_Toc8848568)

[4.6.7 tuya\_ipc\_ss\_stop\_event 78](#_Toc8848569)

[4.6.8 tuya\_ipc\_ss\_delete\_oldest\_event 79](#_Toc8848570)

[4.6.9 tuya\_ipc\_ss\_delete\_all\_files 80](#_Toc8848571)

[4.6.10 tuya\_ipc\_ss\_get\_status 80](#_Toc8848572)

[4.6.11 tuya\_ipc\_pb\_query\_by\_month 81](#_Toc8848573)

[4.6.12 tuya\_ipc\_pb\_query\_by\_day 82](#_Toc8848574)

[4.6.13 tuya\_ipc\_pb\_query\_free\_ts\_arr 83](#_Toc8848575)

[4.6.14 SS\_PB\_EVENT\_CB 84](#_Toc8848576)

[4.6.15 SS\_PB\_GET\_MEDIA\_CB 84](#_Toc8848577)

[4.6.16 tuya\_ipc\_ss\_pb\_start 85](#_Toc8848578)

[4.6.17 tuya\_ipc\_ss\_pb\_set\_status 86](#_Toc8848579)

[4.6.18 tuya\_ipc\_ss\_pb\_stop 87](#_Toc8848580)

[4.6.19 tuya\_ipc\_ss\_pb\_seek 88](#_Toc8848581)

[4.6.20 tuya\_ipc\_ss\_pb\_stop\_all 88](#_Toc8848582)

[4.7 Audio and Video Interface Definitions 89](#_Toc8848583)

[4.7.1 tuya\_ipc\_ring\_buffer\_init 89](#_Toc8848584)

[4.7.2 tuya\_ipc\_ring\_buffer\_append\_data 91](#_Toc8848585)

[4.7.3 tuya\_ipc\_ring\_buffer\_append\_extra\_data 92](#_Toc8848586)

[4.7.4 tuya\_ipc\_ring\_buffer\_get\_video\_frame 93](#_Toc8848587)

[4.7.5 tuya\_ipc\_ring\_buffer\_get\_audio\_frame 95](#_Toc8848588)

[4.7.6 tuya\_ipc\_ring\_buffer\_find\_pre\_video\_by\_frame 96](#_Toc8848589)

[4.7.7 tuya\_ipc\_ring\_buffer\_find\_pre\_audio\_by\_frame 96](#_Toc8848590)

[4.7.8 tuya\_ipc\_ring\_buffer\_get\_pre\_video\_frame 97](#_Toc8848591)

[4.7.9 tuya\_ipc\_ring\_buffer\_get\_pre\_audio\_frame 98](#_Toc8848592)

[4.7.10 tuya\_ipc\_ring\_buffer\_anchor\_user\_to\_node 99](#_Toc8848593)

[4.7.11 tuya\_ipc\_ring\_buffer\_get\_next\_target\_frame 100](#_Toc8848594)

[4.7.12 tuya\_ipc\_ring\_buffer\_clean\_user\_state 101](#_Toc8848595)

[4.7.13 FUNC\_REQUEST\_I\_FRAME 102](#_Toc8848596)

[4.8 PTZ Functions 103](#_Toc8848597)

[4.8.1 tuya\_ipc\_preset\_add 103](#_Toc8848598)

[4.8.2 tuya\_ipc\_preset\_add\_pic 104](#_Toc8848599)

[4.8.3 tuya\_ipc\_preset\_del 105](#_Toc8848600)

[4.8.4 tuya\_ipc\_preset\_get 106](#_Toc8848601)

# 1 Macro Definitions

### 1.1.1 Macro Definitions for Device Information

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| IPC\_STORAGE\_PATH\_LEN | 64 | Storage path length |
| IPC\_PRODUCT\_KEY\_LEN | 16 | Product key length |
| IPC\_UUID\_LEN | 25 | UUID length |
| IPC\_AUTH\_KEY\_LEN | 32 | Length of the authkey that a specific Tuya SDK uses |
| IPC\_SW\_VER\_LEN | 10 | Length of the device firmware version number |
| IPC\_SERIAL\_NUM\_LEN | 32 | Length of a device's unique hardware identifier |
| IPC\_FACTORY\_ID\_LEN | 4 | Vendor ID length |
| IPC\_P2P\_ID\_LEN | 25 | P2P ID length |
| NOTIFICATION\_MOTION\_DETECTION | 115 | Intermittent motion detection DP |
| NOTIFICATION\_CONTENT\_SIZE\_MAX | 150\*1024 | Maximum resolution of reported content |

### 1.1.2 Error Code Macro Definitions

| **Name** | **Value** | **Description** |
| --- | --- | --- |
| OPRT\_OK | (0) | Success |
| OPRT\_COM\_ERROR | (-1) | Common error |
| OPRT\_INVALID\_PARM | (-2) | Invalid parameter |
| OPRT\_MALLOC\_FAILED | (-3) | Memory allocation failure |
| OPRT\_NOT\_SUPPORTED | (-4) | Unsupported function |
| OPRT\_NETWORK\_ERROR | (-5) | Network exception |
|  |  |  |
| OPRT\_INIT\_MUTEX\_ATTR\_FAILED | (-101) | Failed to initialize mutex attributes |
| OPRT\_SET\_MUTEX\_ATTR\_FAILED | (-102) | Failed to set mutex attributes |
| OPRT\_DESTROY\_MUTEX\_ATTR\_FAILED | (-103) | Failed to cancel mutex attributes |
| OPRT\_INIT\_MUTEX\_FAILED | (-104) | Mutex initialization failure |
| OPRT\_MUTEX\_LOCK\_FAILED | (-105) | Mutex lock failure |
| OPRT\_MUTEX\_TRYLOCK\_FAILED | (-106) | Trylock failure |
| OPRT\_MUTEX\_LOCK\_BUSY | (-107) | Busy mutex |
| OPRT\_MUTEX\_UNLOCK\_FAILED | (-108) | Mutex unlock failure |
| OPRT\_MUTEX\_RELEASE\_FAILED | (-109) | Mutex release failure |
| OPRT\_CR\_MUTEX\_ERR | (-110) |  |
| OPRT\_MEM\_PARTITION\_EMPTY | (-111) | Empty memory |
| OPRT\_MEM\_PARTITION\_FULL | (-112) | Full memory |
| OPRT\_MEM\_PARTITION\_NOT\_FOUND | (-113) | No storage area found |
| OPRT\_DONOT\_FOUND\_MODULE | (-114) | No module found |
|  |  |  |
| OPRT\_INIT\_SEM\_FAILED | (-201) |  |
| OPRT\_WAIT\_SEM\_FAILED | (-202) |  |
| OPRT\_POST\_SEM\_FAILED | (-203) |  |
|  |  |  |
| OPRT\_THRD\_STA\_UNVALID | (-301) |  |
| OPRT\_THRD\_CR\_FAILED | (-302) |  |
| OPRT\_THRD\_JOIN\_FAILED | (-303) |  |
| OPRT\_THRD\_SELF\_CAN\_NOT\_JOIN | (-304) |  |
|  |  |  |
| OPRT\_TIMERID\_EXIST | (-401) |  |
| OPRT\_TIMERID\_NOT\_FOUND | (-402) |  |
| OPRT\_TIMERID\_UNVALID | (-403) |  |
| OPRT\_GET\_IDLE\_TIMERID\_ERROR | (-404) |  |
|  |  |  |
| OPRT\_MSG\_NOT\_FOUND | (-501) |  |
| OPRT\_MSG\_LIST\_EMPTY | (-502) |  |
|  |  |  |
| OPRT\_WIFI\_SCAN\_FAIL | (-601) |  |
| OPRT\_WF\_MAC\_SET\_FAIL | (-602) |  |
| OPRT\_WF\_CONN\_FAIL | (-603) |  |
| OPRT\_WF\_NW\_CFG\_FAIL | (-604) |  |
| OPRT\_WF\_AP\_SACN\_FAIL | (-605) |  |
| OPRT\_WF\_NOT\_FIND\_ASS\_AP | (-606) |  |
|  |  |  |
| OPRT\_SOCK\_ERR | (-701) |  |
| OPRT\_SET\_SOCK\_ERR | (-702) |  |
| OPRT\_SOCK\_CONN\_ERR | (-703) |  |
| OPRT\_BUF\_NOT\_ENOUGH | (-704) |  |
| OPRT\_URL\_PARAM\_OUT\_LIMIT | (-705) |  |
| OPRT\_HTTP\_OS\_ERROR | (-706) |  |
| OPRT\_HTTP\_PR\_REQ\_ERROR | (-707) |  |
| OPRT\_HTTP\_SD\_REQ\_ERROR | (-708) |  |
| OPRT\_HTTP\_RD\_ERROR | (-709) |  |
| OPRT\_HTTP\_AD\_HD\_ERROR | (-710) |  |
| OPRT\_HTTP\_GET\_RESP\_ERROR | (-711) |  |
| OPRT\_HTTP\_AES\_INIT\_ERR | (-712) |  |
| OPRT\_HTTP\_AES\_OPEN\_ERR | (-713) |  |
| OPRT\_HTTP\_AES\_SET\_KEY\_ERR | (-714) |  |
| OPRT\_HTTP\_AES\_ENCRYPT\_ERR | (-715) |  |
| OPRT\_CR\_HTTP\_URL\_H\_ERR | (-716) |  |
| OPRT\_HTTPS\_HANDLE\_FAIL | (-717) |  |
| OPRT\_HTTPS\_RESP\_UNVALID | (-718) |  |
| OPRT\_HTTPS\_NO\_SUPPORT\_RANGE | (-719) |  |
|  |  |  |
| OPRT\_CR\_CJSON\_ERR | (-801) |  |
| OPRT\_CJSON\_PARSE\_ERR | (-802) |  |
| OPRT\_CJSON\_GET\_ERR | (-803) |  |
|  |  |  |
| OPRT\_NOT\_FOUND | (-901) |  |
| OPRT\_DP\_ATTR\_ILLEGAL | (-902) |  |
| OPRT\_DP\_TYPE\_PROP\_ILLEGAL | (-903) |  |
| OPRT\_DP\_REPORT\_CLOUD\_ERR | (-904) |  |
| OPRT\_NO\_NEED\_SET\_PRODINFO | (-905) |  |
| OPRT\_NW\_INVALID | (-906) |  |
| OPRT\_SELECT\_ERR | (-907) |  |
| OPRT\_SELECT\_TM | (-908) |  |
| OPRT\_SEND\_ERR | (-909) |  |
| OPRT\_DEV\_NOT\_BIND | (-910) |  |
| OPRT\_FW\_UG\_FAILED | (-911) |  |
| OPRT\_VER\_FMT\_ERR | (-912) |  |
| OPRT\_FW\_NOT\_EXIST | (-913) |  |
| OPRT\_SEM\_CR\_ERR | (-914) |  |
| OPRT\_SELECT\_TIMEOUT | (-915) |  |
| OPRT\_GW\_MQ\_OFFLILNE | (-916) |  |
| OPRT\_NOT\_SPT\_CLX\_DP | (-917) |  |
| OPRT\_RECV\_ERR | (-918) |  |
| OPRT\_UG\_PKG\_NOT\_ENOUGH | (-919) |  |
| OPRT\_SCMA\_INVALID | (-920) |  |
| OPRT\_PRODECT\_KEY\_NULL | (-921) |  |
| OPRT\_DEVICE\_VER\_NULL | (-922) |  |
| OPRT\_MSG\_OUT\_OF\_LMT | (-923) |  |
| OPRT\_NOT\_FOUND\_AUTH\_SSID | (-924) |  |
| OPRT\_SOCKET\_FAULT | (-925) |  |
| OPRT\_MQ\_PUBLISH\_TIMEOUT | (-926) |  |
| OPRT\_GW\_NOT\_EXIST | (-927) |  |
| OPRT\_GW\_SCHEMA\_SIZE\_LMT\_OUT | (-928) |  |
| OPRT\_DEV\_DP\_CNT\_INVALID | (-929) |  |
| OPRT\_TOKEN\_OVERTIME | (-930) |  |
| OPRT\_WF\_NW\_CFG\_RECV\_CONTINUE | (-931) |  |
| OPRT\_RECV\_DA\_NOT\_ENOUGH | (-932) |  |
| OPRT\_SERV\_VRFY\_FAIL | (-933) |  |
| OPRT\_KVS\_WR\_FAIL | (-934) |  |
| OPRT\_KVS\_RD\_FAIL | (-935) |  |
| OPRT\_NO\_AUTHENTICATION | (-936) |  |
| OPRT\_CMD\_OUT\_OF\_TIME\_NOT\_EXEC | (-937) |  |
| OPRT\_CMD\_NOT\_EXEC | (-938) |  |
| OPRT\_CRC32\_FAILED | (-939) |  |
| OPRT\_LAN\_NO\_CLIENT | (-940) |  |
| OPRT\_WF\_INTF\_LACK\_REG\_CB | (-941) |  |
| OPRT\_WF\_INTF\_NO\_REG\_CB | (-942) |  |
| OPRT\_STORAGE\_UPLOAD | (-943) |  |
| OPRT\_DP\_ID\_NOT\_FOUND | (-944) | Invalid DP |
| OPRT\_DP\_TP\_NOT\_MATCH | (-945) |  |
| OPRT\_DEV\_NEED\_REGISTER | (-946) |  |
| OPRT\_DEV\_RESET\_FACTORY | (-947) | Factory defaults restoration |
| OPRT\_API\_VERSION\_WRONG | (-948) |  |
|  |  |  |
| OPRT\_OPEN\_FILE\_FAILED | (-1001) |  |
| OPRT\_PRINT\_LOG\_LEVEL\_HIGHER | (-1002) |  |
| OPRT\_FORMAT\_STRING\_FAILED | (-1003) |  |
| OPRT\_STRING\_TOO\_LONG | (-1004) |  |
| OPRT\_WR\_FLASH\_ERROR | (-1005) |  |
|  |  |  |
| OPRT\_ROUTER\_NOT\_FIND | (-1006) |  |
| OPRT\_EXCEED\_UPPER\_LIMIT | (-1007) |  |
| OPRT\_SECURITY\_VERIFY\_NOT\_OK | (-1008) |  |
|  |  |  |
| OPRT\_OSS\_OPERATION\_INIT\_FAILED | (-1100) |  |
| OPRT\_OSS\_OPERATION\_UPDATE\_FAILED | (-1101) |  |
| OPRT\_OSS\_OPERATION\_UPLOAD\_FAILED | (-1102) |  |
| OPRT\_MOTION\_DETECTION\_ALARM\_PACK\_REPORT\_FAILED | (-1103) |  |
| OPRT\_TRANSFER\_INIT\_FAILED | (-1104) |  |
|  |  |  |
| OPRT\_EVENT\_STATUS\_INVALID | (-2000) | Invalid event status |
| OPRT\_ENCRYPT\_KEY\_UPDATED | (-2001) |  |
| OPRT\_ORDER\_EXPIRE | (-2002) |  |
| OPRT\_NO\_MORE\_DATA | (-2003) |  |
| OPRT\_CLOUD\_STORAGE\_OFF | (-2004) |  |
| OPRT\_CLOUD\_STORAGE\_STATUS\_INVALID | (-2005) |  |
| OPRT\_REMOTE\_API\_RUN\_UNKNOW\_FAILED | (-2006) | Abnormal server interface |
| OPRT\_DEVICE\_REMOVED | (-2007) |  |

### 1.1.3 Constant Macro Definitions

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| ENCRYPT\_KEY\_LEN | 16 | Maximum storage encryption key length |
| GW\_UUID\_LEN | 25 | Maximum UUID length |
| AUTH\_KEY\_LEN | 32 | Maximum authkey length |
| PRODUCT\_KEY\_LEN | 16 | Maximum product key length |
| SW\_VER\_LEN | 10 | Maximum version string length |
|  |  |  |
| GW\_ID\_LEN | 25 | Maximum gateway ID length |
| DEV\_ID\_LEN | 25 | Maximum device ID length |
| SCHEMA\_ID\_LEN | 16 |  |
| BIND\_TOKEN\_LEN | 16 | Maximum bound token length |
| TOKEN\_LEN | 8 | Maximum token length |
| REGION\_LEN | 2 | Region length |
| REGIST\_KEY\_LEN | 4 | Registration key length |
| HTTP\_URL\_LMT | 128 | Maximum URL length |
| MQ\_URL\_LMT | 128 | Maximum MQTT URL length |
| TIME\_ZONE\_LEN | 10 | Maximum time zone length |
| SUMMER\_TM\_ZONE\_LEN | 256 | Maximum length of the daylight saving time string |
|  |  |  |
| SEC\_KEY\_LEN | 16 | Maximum security key string length |
| LOCAL\_KEY\_LEN | 16 | Maximum the local key string length |
|  |  |  |
| WXAPPID\_LEN | 32 | WX app ID length |
| WXUUID\_LEN | 32 | WX UUID length |
|  |  |  |
| GRP\_ID\_LEN | 5 | GRP ID length |
| SCENE\_ID\_LEN | 3 | Maximum scenario ID length |
| LC\_GW\_SLEEP\_HB\_LMT\_S | 12\*3600 | Maximum gateway sleep duration |

### 1.1.4 Macro Definitions of the Group and Scenario Error Code

| **Name** | **Value** | **Description** |
| --- | --- | --- |
| M\_SCE\_SUCCESS | 0 | Success |
| M\_SCE\_ERR\_EXCEED | 1 | Insufficient space |
| M\_SCE\_ERR\_TIMEOUT | 2 | Communication timeout |
| M\_SCE\_ERR\_PARAM | 3 | Parameter out of range |
| M\_SCE\_WRITE\_FILE | 4 | Database file writing error |
| M\_SCE\_ERR\_OTHER | 5 | Other errors |

### 1.1.5 DP Message Macro Definitions

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| **Basic Function Settings App Page** | | |
| #define TUYA\_DP\_SLEEP\_MODE | 105 | Sleep mode, Boolean. The value can be: **true**: sleep **false**: not sleep |
| #define TUYA\_DP\_LIGHT | 101 | Status indicator, Boolean. The value can be: **true**: on **false**: off |
| #define TUYA\_DP\_FLIP | 103 | Recording screen flip, Boolean. The value can be: **true**: flip **false**: normal |
| #define TUYA\_DP\_WATERMARK | 104 | Video watermark, Boolean. The value can be: **true**: enable **false**: disable |
| #define TUYA\_DP\_WDR | 107 | Wide dynamic range mode, Boolean. The value can be: **true**: enable **false**: disable |
| #define TUYA\_DP\_NIGHT\_MODE | 108 | Infrared (IR) night vision function, enum. The value can be: **0**: auto mode **1:** disable **2**: enable |
| #define TUYA\_DP\_CALIBRATION | 132 | Camera calibration, Boolean. |
| **Motion Detection Alarm App Page** | | |
| #define TUYA\_DP\_ALARM\_FUNCTION | 134 | Motion detection alarm function, Boolean. The value can be: **true**: on **false**: off |
| #define TUYA\_DP\_ALARM\_SENSITIVITY | 106 | Motion detection alarm sensitivity, enum. The value can be: **0**: low sensitivity **1**: medium sensitivity **2:** high sensitivity |
| **SD Card Settings App Page** | | |
| #define TUYA\_DP\_SD\_STATUS\_ONLY\_GET | 110 | SD card status, value type. The value can be: **1**: normal **2**: abnormal **3**: insufficient space **4**: formatting **5**: no SD card |
| #define TUYA\_DP\_SD\_STORAGE\_ONLY\_GET | 109 | SD card capacity, string. The value contains the total capacity, current usage, and remaining capacity, in kilobytes (KB). |
| #define TUYA\_DP\_SD\_RECORD\_ENABLE | 150 | SD card recording function, Boolean. The value can be: **true**: enable **false**: disable |
| #define TUYA\_DP\_SD\_RECORD\_MODE | 151 | SD card recording mode selection, enum. The value can be: **0**: event recording **1:** continuous recording |
| #define TUYA\_DP\_SD\_UMOUNT | 112 | SD card unmounting, Boolean. The value can be: **true**: unmounted **false**: mounted |
| #define TUYA\_DP\_SD\_FORMAT | 111 | SD card formatting, Boolean. |
| #define TUYA\_DP\_SD\_FORMAT\_STATUS\_ONLY\_ GET | 117 | SD card formatting status, value type. The value can be: –**2000:** formatting –**2001**: formatting exceptions –**2002**: no SD card –**2003**: SD card error. Positive value: formatting progress |
| **Camera Control App Page** | | |
| #define TUYA\_DP\_PTZ\_CONTROL | 119 | PTZ rotation direction, enum. The value can be: **0**: up **1**: up right **2**: right **3**: down right **4**: down **5**: down left **6**: left **7**: up left  PTZ rotation direction (from SDK 4.0), enum. The value can be: **0**: up right **1**: right **2**: down right **3**: down **4**: down left **5**: left **6**: up left **7**: up |
| #define TUYA\_DP\_PTZ\_STOP | 116 | PTZ rotation stop, Boolean. |
| #define TUYA\_DP\_PTZ\_CHECK | 132 | PTZ self-check. The value can be: **0**: disable **1**: enable The default value is 0. |
| #define TUYA\_DP\_TRACK\_ENABLE | 161 | Movement tracking, Boolean. The value can be: **true**: enable **false**: disable |
| #define TUYA\_DP\_HUM\_FILTER | 170 | Human filtering, Boolean. The value can be: **true**: enable **false**: disable The default value is false. |
| #define TUYA\_DP\_PATROL\_SWITCH | 174 | Patrol switch. The value can be: **0**: disable **1**: enable |
| #define TUYA\_DP\_PATROL\_MODE | 175 | Patrol type mode. The value can be: **0**: panoramic patrol **1**: preset patrol |
| #define TUYA\_DP\_PATROL\_TMODE | 176 | Patrol time mode. The value can be: **0**: all-day patrol **1**: timed patrol |
| #define TUYA\_DP\_PATROL\_TIME | 177 | Patrol time. |
| #define TUYA\_DP\_PRESET\_SET | 178 | Preset operation, string. The value **type:1** indicates adding, and **type: 2** indicates deleting. Different data types have different strings. |
| #define TUYA\_DP\_PATROL\_STATE | 179 | Patrol status query command. The returned value can be: **0**: panoramic patrol **1**: preset patrol **2**: non-patrol mode |
|  |  |  |
| **Motion Detection Zone Function** | | |
| #define TUYA\_DP\_ALARM\_ZONE\_ENABLE | 168 | Detection zone, Boolean. The value can be: **true**: enable **false**: disable |
| #define TUYA\_DP\_ALARM\_ZONE\_DRAW | 169 | Detection zone setting, string, for example, {"num":1, "region0":{"x":30,"y":40,"xlen":50,"ylen":60}} |
| #define TUYA\_DP\_DOOR\_BELL | 136 | Doorbell call, string, for example, "current timestamp" |
| #define TUYA\_DP\_BLUB\_SWITCH | 138 | Special light control, Boolean. The value can be: **true**: enable **false**: disable |
| #define TUYA\_DP\_SOUND\_DETECT | 139 | Decibel detection, Boolean. The value can be: **true**: enable **false**: disable |
| #define TUYA\_DP\_SOUND\_SENSITIVITY | 140 | Decibel detection sensitivity, enum. The value can be: **0**: low sensitivity **1**: high sensitivity |
| #define TUYA\_DP\_SOUND\_ALARM | 141 | Decibel alarm channel, string, for example, "current timestamp" |
| #define TUYA\_DP\_TEMPERATURE | 142 | Temperature detection, value type. The value range is from 0 to 50. |
| #define TUYA\_DP\_HUMIDITY | 143 | Humidity detection, value type. The value range is from 0 to 100. |
| #define TUYA\_DP\_ELECTRICITY | 145 | Battery percentage, value type. The value range is from 0 to 100. |
| #define TUYA\_DP\_POWERMODE | 146 | Power supply mode, enum. The value can be: **0:** battery power supply **1**: mains power supply or battery charging |
| #define TUYA\_DP\_LOWELECTRIC | 147 | Percentage-based low battery alarm threshold, value type |
| #define TUYA\_DP\_DOOR\_STATUS | 149 | Doorbell status notification, Boolean. The value can be: **true**: enable **false**: disable |
| #define TUYA\_DP\_MOTION\_DETECTION\_ALARM | 115 | Motion detection message alarm |
| #define TUYA\_DP\_DOOR\_BELL\_SNAP | 154 | Doorbell push screenshot tips |

### 1.1.6 Other Macro Definitions

| **Name** | **Value** | **Description** |
| --- | --- | --- |
| GW\_VIRTUAL | 0x00 | Product that only supports device functions and not sub-device functions, for example, the SoC or MCU |
| GW\_ENTITY\_ATTH\_DEV | 0x01 | Product that has both sub-device and device functions |
| GW\_ENTITY | 0x02 | Product that only has sub-device functions |
|  |  |  |
| GNS\_UNVALID | 0 | Device disconnected from LAN and WAN |
| GNS\_LAN\_VALID | 1 | Device connected to LAN but disconnected from WAN |
| GNS\_WAN\_VALID | 2 | Device connected to LAN and WAN |
|  |  |  |
| EXT\_UNAUTH | 0 |  |
| EXT\_PROD\_TEST | 1 |  |
| EXT\_NORMAL\_S | 2 |  |
| EXT\_GW\_UPGRD | 3 |  |
| EXT\_DEV\_UPGRD | 4 |  |
|  |  |  |
| DEV\_NM\_ATH\_SNGL | 0 | NetLink module with a single device (devid == gwid) |
| DEV\_NM\_NOT\_ATH\_SNGL | 1 | NetLink module without a single device (devid == gwid) |
| DEV\_ZB\_SNGL | 2 | ZigBee device |
| DEV\_RF433\_SNGL | 3 | RF433 device |
| DEV\_BLE\_SNGL | 4 | Bluetooth Low Energy (BLE) device |
|  |  |  |
| GP\_DEV\_ZB | DEV\_ZB\_SNGL | ZigBee device |
| GP\_DEV\_RF433 | DEV\_RF433\_SNGL | RF433 device |
| GP\_DEV\_BLE | DEV\_BLE\_SNGL | BLE device |
|  |  |  |
| GW\_ATTACH\_ATTR\_LMT | 4 |  |
|  |  |  |
| T\_OBJ | 0 | dp is value,str,enum,bool,bitmap |
| T\_RAW | 1 | raw type |
| T\_FILE | 2 | file type |
|  |  |  |
| M\_RW | 0 | Readable and writable DP on the cloud or in the app |
| M\_WR | 1 | Write-only DP on the cloud or in the app |
| M\_RO | 2 | Read-only DP on the cloud or in the app |
|  |  |  |
| PROP\_BOOL | 0 | Boolean |
| PROP\_VALUE | 1 | Value |
| PROP\_STR | 2 | String |
| PROP\_ENUM | 3 | Enum |
| PROP\_BITMAP | 4 | Bitmap |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 2 Enum Definitions

### 2.1.1 Types of Uploaded Notification Content

typedef enum{

NOTIFICATION\_CONTENT\_MP4 = 0,

NOTIFICATION\_CONTENT\_JPEG,

NOTIFICATION\_CONTENT\_PNG,

NOTIFICATION\_CONTENT\_MAX,

} NOTIFICATION\_CONTENT\_TYPE\_E;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| NOTIFICATION\_CONTENT\_MP4 | 0 | Video, MP4 file |
| NOTIFICATION\_CONTENT\_JPEG | 1 | Image, JPEG file |
| NOTIFICATION\_CONTENT\_PNG | 2 | Image, PNG file |
| NOTIFICATION\_CONTENT\_MAX | 3 | Maximum number of enum values (excluding **3**) |

### 2.1.2 IPC Registration States

typedef enum

{

E\_IPC\_UNREGISTERED = 0,

E\_IPC\_REGISTERED = 1,

E\_IPC\_ACTIVEATED = 2

}IPC\_REGISTER\_STATUS;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| E\_IPC\_UNREGISTERED | 0 | Unregistered and unactivated |
| E\_IPC\_REGISTERED | 1 | Intermediate state |
| E\_IPC\_ACTIVEATED | 2 | Activated and bound state, which prevails |

### 2.1.3 **Gateway States**

typedef enum

{

IPC\_GW\_UNREGISTERED = 0,

IPC\_GW\_REGISTERED,

IPC\_GW\_ACTIVED,

IPC\_GW\_MQTT\_ONLINE,

IPC\_GW\_MQTT\_OFFLINE,

}IPC\_GW\_STATUS\_E;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| IPC\_GW\_UNREGISTERED | 0 | Unregistered |
| IPC\_GW\_REGISTERED | 1 | Unactivated |
| IPC\_GW\_ACTIVED | 2 | Activated |
| IPC\_GW\_MQTT\_ONLINE | 3 | Connected to the MQTT |
| IPC\_GW\_MQTT\_OFFLINE | 4 | Disconnected from the MQTT |

### 2.1.4 Input Types of Direct Network Access

typedef enum{

TUYA\_IPC\_DIRECT\_CONNECT\_QRCODE = 0,

TUYA\_IPC\_DIRECT\_CONNECT\_MAX,

} TUYA\_IPC\_DIRECT\_CONNECT\_TYPE\_E;

|  |  |  |
| --- | --- | --- |
| **Enum** **Name** | **Enum Value** | **Description** |
| TUYA\_IPC\_DIRECT\_CONNECT\_QRCODE | 0 | QR code net-pairing |
| TUYA\_IPC\_DIRECT\_CONNECT\_MAX | 1 |  |

### 2.2.1 Video Code Rate

typedef enum

{

TUYA\_VIDEO\_BITRATE\_64K = 64,

TUYA\_VIDEO\_BITRATE\_128K = 128,

TUYA\_VIDEO\_BITRATE\_256K = 256,

TUYA\_VIDEO\_BITRATE\_512K = 512,

TUYA\_VIDEO\_BITRATE\_768K = 768,

TUYA\_VIDEO\_BITRATE\_1M = 1024,

TUYA\_VIDEO\_BITRATE\_1\_5M = 1536

}TUYA\_VIDEO\_BITRATE\_E;

|  |  |  |
| --- | --- | --- |
| **Enum** **Name** | **Enum** **Value** | **Description** |
| TUYA\_VIDEO\_BITRATE\_64K | 64 | The code rate is 64 kbits/s. |
| TUYA\_VIDEO\_BITRATE\_128K | 128, | The code rate is 128 kbits/s. |
| TUYA\_VIDEO\_BITRATE\_256K | 256, | The code rate is 256 kbits/s. |
| TUYA\_VIDEO\_BITRATE\_512K | 512, | The code rate is 512 kbits/s. |
| TUYA\_VIDEO\_BITRATE\_768K | 768, | The code rate is 768 kbits/s. |
| TUYA\_VIDEO\_BITRATE\_1M | 1024, | The code rate is 1024 kbits/s. |
| TUYA\_VIDEO\_BITRATE\_1\_5M | 1536 | The code rate is 1536 kbits/s. |

### 2.2.2 Audio and Video Coding Protocols

typedef enum

{

TUYA\_CODEC\_VIDEO\_MPEG4 = 0,

TUYA\_CODEC\_VIDEO\_H263,

TUYA\_CODEC\_VIDEO\_H264,

TUYA\_CODEC\_VIDEO\_MJPEG,

TUYA\_CODEC\_VIDEO\_H265,

TUYA\_CODEC\_VIDEO\_MAX = 99,

TUYA\_CODEC\_AUDIO\_ADPCM,

TUYA\_CODEC\_AUDIO\_PCM,

TUYA\_CODEC\_AUDIO\_AAC\_RAW,

TUYA\_CODEC\_AUDIO\_AAC\_ADTS,

TUYA\_CODEC\_AUDIO\_AAC\_LATM,

TUYA\_CODEC\_AUDIO\_G711U,

TUYA\_CODEC\_AUDIO\_G711A,

TUYA\_CODEC\_AUDIO\_G726,

TUYA\_CODEC\_AUDIO\_SPEEX,

TUYA\_CODEC\_AUDIO\_MP3,

TUYA\_CODEC\_AUDIO\_MAX = 199,

TUYA\_CODEC\_INVALID

}TUYA\_CODEC\_ID;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| TUYA\_CODEC\_VIDEO\_MPEG4 | 0 | MPEG-4 |
| TUYA\_CODEC\_VIDEO\_H263 | 1 | H.263 |
| TUYA\_CODEC\_VIDEO\_H264 | 2 | H.264 |
| TUYA\_CODEC\_VIDEO\_MJPEG | 3 | MJPEG |
| TUYA\_CODEC\_VIDEO\_H265 | 4 | H.265 |
| TUYA\_CODEC\_VIDEO\_MAX | 99 | Maximum number of enum values |
| TUYA\_CODEC\_AUDIO\_ADPCM | 100 | Adaptive differential pulse-code modulation (ADPCM) |
| TUYA\_CODEC\_AUDIO\_PCM | 101 | Pulse-code modulation (PCM) |
| TUYA\_CODEC\_AUDIO\_AAC\_RAW | 102 | AA\_RAW |
| TUYA\_CODEC\_AUDIO\_AAC\_ADTS | 103 | AAC\_ADTS |
| TUYA\_CODEC\_AUDIO\_AAC\_LATM | 104 | AAC\_LATM |
| TUYA\_CODEC\_AUDIO\_G711U | 105 | G711U |
| TUYA\_CODEC\_AUDIO\_G711A | 106 | G711A |
| TUYA\_CODEC\_AUDIO\_G726 | 107 | G.726 |
| TUYA\_CODEC\_AUDIO\_SPEEX | 108 | Speex |
| TUYA\_CODEC\_AUDIO\_MP3 | 109 | MP3 |
| TUYA\_CODEC\_AUDIO\_MAX | 199 | Maximum number of enum values |
| TUYA\_CODEC\_INVALID | 200 | Invalid protocol |

### 2.2.3 Audio Sampling Rate

typedef enum

{

TUYA\_AUDIO\_SAMPLE\_8K = 8000,

TUYA\_AUDIO\_SAMPLE\_11K = 11000,

TUYA\_AUDIO\_SAMPLE\_12K = 12000,

TUYA\_AUDIO\_SAMPLE\_16K = 16000,

TUYA\_AUDIO\_SAMPLE\_22K = 22000,

TUYA\_AUDIO\_SAMPLE\_24K = 24000,

TUYA\_AUDIO\_SAMPLE\_32K = 32000,

TUYA\_AUDIO\_SAMPLE\_44K = 44000,

TUYA\_AUDIO\_SAMPLE\_48K = 48000,

TUYA\_AUDIO\_SAMPLE\_MAX = 0xFFFFFFFF

}TUYA\_AUDIO\_SAMPLE\_E;

| **Enum Name** | **Enum Value** | **Description** |
| --- | --- | --- |
| TUYA\_AUDIO\_SAMPLE\_8K | 8000 | The sampling rate is 8 kHz. |
| TUYA\_AUDIO\_SAMPLE\_11K | 11000 | The sampling rate is 11 kHz. |
| TUYA\_AUDIO\_SAMPLE\_12K | 12000 | The sampling rate is 12 kHz. |
| TUYA\_AUDIO\_SAMPLE\_16K | 16000 | The sampling rate is 16 kHz. |
| TUYA\_AUDIO\_SAMPLE\_22K | 22000 | The sampling rate is 22 kHz. |
| TUYA\_AUDIO\_SAMPLE\_24K | 24000 | The sampling rate is 24 kHz. |
| TUYA\_AUDIO\_SAMPLE\_32K | 32000 | The sampling rate is 32 kHz. |
| TUYA\_AUDIO\_SAMPLE\_44K | 44000 | The sampling rate is 44 kHz. |
| TUYA\_AUDIO\_SAMPLE\_48K | 48000 | The sampling rate is 48 kHz. |
| TUYA\_AUDIO\_SAMPLE\_MAX | 0xFFFFFFFF |  |

### 2.2.4 Audio Bits

typedef enum

{

TUYA\_AUDIO\_DATABITS\_8 = 8,

TUYA\_AUDIO\_DATABITS\_16 = 16,

TUYA\_AUDIO\_DATABITS\_MAX = 0xFF

}TUYA\_AUDIO\_DATABITS\_E;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| TUYA\_AUDIO\_DATABITS\_8 | 8 | 8 bits |
| TUYA\_AUDIO\_DATABITS\_16 | 16 | 16 bits |
| TUYA\_AUDIO\_DATABITS\_MAX | 0xFF | 255 bits |

### 2.2.5 Audio Channels

typedef enum

{

TUYA\_AUDIO\_CHANNEL\_MONO,

TUYA\_AUDIO\_CHANNEL\_STERO,

}TUYA\_AUDIO\_CHANNEL\_E;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| TUYA\_AUDIO\_CHANNEL\_MONO | 0 | Mono |
| TUYA\_AUDIO\_CHANNEL\_STERO | 1 | Stereo |

### 2.2.6 Cloud Storage Order Types

typedef enum

{

ClOUD\_STORAGE\_TYPE\_CONTINUE,

ClOUD\_STORAGE\_TYPE\_EVENT,

ClOUD\_STORAGE\_TYPE\_INVALID

}ClOUD\_STORAGE\_TYPE\_E;

|  |  |  |
| --- | --- | --- |
| **Enum** **Name** | **Enum** **Value** | **Description** |
| ClOUD\_STORAGE\_TYPE\_CONTINUE | 0 | Data will be uploaded continuously to the cloud for storage until the end time of the order. |
| ClOUD\_STORAGE\_TYPE\_EVENT | 1 | Only event data will be uploaded to the cloud for storage. The start and end are triggered by user-defined data. |
| ClOUD\_STORAGE\_TYPE\_INVALID | 2 | No order or other abnormal status. |

### 2.2.7 Event Types

typedef enum

{

EVENT\_TYPE\_MOTION\_DETECT,

EVENT\_TYPE\_DOOR\_BELL,

EVENT\_TYPE\_INVALID

}ClOUD\_STORAGE\_EVENT\_TYPE\_E;

|  |  |  |
| --- | --- | --- |
| **Enum** **Name** | **Enum** **Value** | **Description** |
| EVENT\_TYPE\_MOTION\_DETECT | 0 | Storage triggered by motion detection |
| EVENT\_TYPE\_DOOR\_BELL | 1 | Storage triggered by doorbell wakeup |
| EVENT\_TYPE\_INVALID | 2 | Invalid event |

### 2.2.8 Event States

typedef enum

{

EVENT\_NONE,

EVENT\_ONGOING,

EVENT\_INVALID

}EVENT\_STATUS\_E;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| EVENT\_NONE | 0 | No event |
| EVENT\_ONGOING | 1 | Ongoing |
| EVENT\_INVALID | 2 | Invalid state |

### 2.3.1 Writing Files to an SD Card

typedef enum {

SS\_WRITE\_MODE\_NONE,

SS\_WRITE\_MODE\_EVENT,

SS\_WRITE\_MODE\_ALL,

SS\_WRITE\_MODE\_MAX

}STREAM\_STORAGE\_WRITE\_MODE\_E;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| SS\_WRITE\_MODE\_NONE | 0 | No file is written to the SD card. |
| SS\_WRITE\_MODE\_EVENT | 1 | Files are written to the SD card when events occur. |
| SS\_WRITE\_MODE\_ALL | 2 | Files are always written to the SD card. |
| SS\_WRITE\_MODE\_MAX | 3 |  |

### 2.3.2 Deleting Files

typedef enum

{

E\_DELETE\_NONE,

E\_DELETE\_YEAR,

E\_DELETE\_MONTH,

E\_DELETE\_DAY,

E\_DELETE\_EVENT,

E\_DELETE\_INVALID,

E\_DELETE\_FS\_ERROR

}DELETE\_LEVEL\_E;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| E\_DELETE\_NONE | 0 | No files are deleted. |
| E\_DELETE\_YEAR | 1 | Files are deleted by year. |
| E\_DELETE\_MONTH | 2 | Files are deleted by month. |
| E\_DELETE\_DAY | 3 | Files are deleted by day. |
| E\_DELETE\_EVENT | 4 | Files are deleted by event. |
| E\_DELETE\_INVALID | 5 |  |
| E\_DELETE\_FS\_ERROR | 6 |  |

### 2.3.3 Storage States

typedef enum

{

E\_STORAGE\_STOP,

E\_STORAGE\_START,

//The following internal status interfaces cannot be set externally.

E\_STORAGE\_ONGOING,

E\_STORAGE\_READY\_TO\_STOP,

E\_STORAGE\_RESTART,

E\_STORAGE\_INVALID

}STORAGE\_STATUS\_E;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| E\_STORAGE\_STOP | 0 | Pause |
| E\_STORAGE\_START | 1 | Start |
| E\_STORAGE\_ONGOING | 2 | Storing |
| E\_STORAGE\_READY\_TO\_STOP | 3 | Ready to pause |
| E\_STORAGE\_RESTART | 4 | Restart |
| E\_STORAGE\_INVALID | 5 | Invalid parameter |

### 2.3.4 Event Playback

typedef enum

{

SS\_PB\_FINISH = 0,

SS\_PB\_NEWFILE,

}SS\_PB\_EVENT\_E;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| SS\_PB\_FINISH | 0 | The current file has finished playing. |
| SS\_PB\_NEWFILE | 1 | There is a continuous file to play. |

### 2.3.5 Frame Types

typedef enum

{

E\_VIDEO\_PB\_FRAME = 0,

E\_VIDEO\_I\_FRAME,

E\_VIDEO\_TS\_FRAME,

E\_AUDIO\_FRAME,

E\_MEDIA\_FRAME\_TYPE\_MAX

}MEDIA\_FRAME\_TYPE\_E;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| E\_VIDEO\_PB\_FRAME | 0 | PG frame |
| E\_VIDEO\_I\_FRAME | 1 | I-Frame |
| E\_VIDEO\_TS\_FRAME | 2 | TS frame |
| E\_AUDIO\_FRAME | 3 | Audio frame |
| E\_MEDIA\_FRAME\_TYPE\_MAX | 4 |  |

### 2.3.6 Playback States

typedef enum

{

SS\_PB\_MUTE,

SS\_PB\_UN\_MUTE,

SS\_PB\_PAUSE,

SS\_PB\_RESUME,

}SS\_PB\_STATUS\_E;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| SS\_PB\_MUTE | 0 | Mute |
| SS\_PB\_UN\_MUTE | 1 | Unmute |
| SS\_PB\_PAUSE | 2 | Pause |
| SS\_PB\_RESUME | 3 | Continue |

### 2.4.1 Reset Types

typedef enum {

GW\_LOCAL\_RESET\_FACTORY = 0,

GW\_REMOTE\_UNACTIVE,

GW\_LOCAL\_UNACTIVE,

GW\_REMOTE\_RESET\_FACTORY

}GW\_RESET\_TYPE\_E;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| GW\_LOCAL\_RESET\_FACTORY | 0 | Local device reset |
| GW\_REMOTE\_UNACTIVE | 1 | Remote inactive reset |
| GW\_LOCAL\_UNACTIVE | 2 | Local inactive reset |
| GW\_REMOTE\_RESET\_FACTORY | 3 | Remote device reset |

### 2.5.1 Maximum Number of Video and Audio Stream Channels

typedef enum

{

E\_CHANNEL\_VIDEO\_MAIN = 0,

E\_CHANNEL\_VIDEO\_SUB,

E\_CHANNEL\_VIDEO\_3RD,

E\_CHANNEL\_VIDEO\_4TH,

E\_CHANNEL\_VIDEO\_MAX = 8, // Maximum number of video stream channels

E\_CHANNEL\_AUDIO,

E\_CHANNEL\_AUDIO\_2RD,

E\_CHANNEL\_AUDIO\_3RD,

E\_CHANNEL\_AUDIO\_4TH,

E\_CHANNEL\_MAX = 16

}CHANNEL\_E;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| E\_CHANNEL\_VIDEO\_MAIN | 0 | Main stream |
| E\_CHANNEL\_VIDEO\_SUB | 1 | Substream 1 |
| E\_CHANNEL\_VIDEO\_3RD | 2 | Substream 2 |
| E\_CHANNEL\_VIDEO\_4TH | 3 | Substream 3 |
| E\_CHANNEL\_VIDEO\_MAX | 8 | Maximum number of video stream channels |
| E\_CHANNEL\_AUDIO | 9 | Main audio stream |
| E\_CHANNEL\_AUDIO\_2RD | 10 | Audio stream 1 |
| E\_CHANNEL\_AUDIO\_3RD | 11 | Audio stream 2 |
| E\_CHANNEL\_AUDIO\_4TH | 12 | Audio stream 3 |
| E\_CHANNEL\_MAX | 16 | Maximum number of audio stream channels |

### 2.5.2 User Index

typedef enum

{

E\_USER\_STREAM\_STORAGE = 0,

E\_USER\_COULD\_STORAGE = 1,

E\_USER\_ECHO\_SHOW = 2,

E\_USER\_CHROMECAST = 3,

E\_USER\_P2P\_USER = 4, //Multi-channel P2P, which uses the E\_USER\_P2P\_USER and P2P index to distinguish users

E\_USER\_RTSP = 10,

E\_USER\_NUM\_MAX = 16

}USER\_INDEX\_E;

|  |  |  |
| --- | --- | --- |
| **Enum Name** | **Enum Value** | **Description** |
| E\_USER\_STREAM\_STORAGE | 0 | Local storage |
| E\_USER\_COULD\_STORAGE | 1 | Cloud storage |
| E\_USER\_ECHO\_SHOW | 2 | Echo Show user |
| E\_USER\_CHROMECAST | 3 | Chromecast user |
| E\_USER\_P2P\_USER | 4 | P2P user |
| E\_USER\_RTSP | 10 | Real Time Streaming Protocol (RTSP) |
| E\_USER\_NUM\_MAX | 16 | Maximum number of users |

# 3 Structure Definitions

### 3.1.1 IPC Environment Variables

typedef struct

{

CHAR\_T storage\_path[IPC\_STORAGE\_PATH\_LEN + 1];

CHAR\_Tproduct\_key[IPC\_PRODUCT\_KEY\_LEN + 1];

CHAR\_T uuid[IPC\_UUID\_LEN + 1];

CHAR\_T auth\_key[IPC\_AUTH\_KEY\_LEN + 1];

CHAR\_T p2p\_id[IPC\_P2P\_ID\_LEN + 1];

CHAR\_T dev\_sw\_version[IPC\_SW\_VER\_LEN + 1];

CHAR\_T dev\_serial\_num[IPC\_SERIAL\_NUM\_LEN + 1];

DEV\_OBJ\_DP\_CMD\_CB dev\_obj\_dp\_cb;

DEV\_DP\_QUERY\_CB ev\_dp\_query\_cb;

GW\_STATUS\_CHANGED\_CB status\_changed\_cb;

GW\_UG\_INFORM\_CB gw\_ug\_cb;

TUYA\_RST\_INFORM\_CB gw\_rst\_cb;

TUYA\_RESTART\_INFORM\_CB w\_restart\_cb;

BOOL\_T mem\_save\_mode;

VOID \*p\_reserved;

}TUYA\_IPC\_ENV\_VAR\_S;

| **Parameter** | **Description** |
| --- | --- |
| storage\_path | Readable and writable storage path of Tuya SDK internal configuration files. Data will not be lost in case of power failure. |
| product\_key | Product key. |
| uuid | UUID for Tuya SDK. If this parameter is empty, the SDK uses the pre-written value in the configuration file. |
| auth\_key | Authkey for Tuya SDK. If this parameter is empty, the SDK uses the pre-written value in the configuration file. |
| p2p\_id | P2P ID for Tuya SDK. If the local user does not have a PIP ID, the value of this parameter is null. |
| dev\_sw\_version | Device firmware version number. |
| dev\_serial\_num | Unique hardware identifier of a device. |
| dev\_obj\_dp\_cb | Tuya command distribution function. |
| dev\_dp\_query\_cb | Tuya query distribution function. |
| status\_changed\_cb | Device status change callback function. The value can be **NULL.** |
| gw\_ug\_cb | Callback function for getting the firmware upgrade information in IPC mode. The value can be **NULL.** |
| gw\_rst\_cb | Callback function for resetting the IPC in IPC mode. The value can be **NULL.** |
| gw\_restart\_cb | Callback function for restarting the IPC progress in IPC mode. The value can be **NULL.** |
| mem\_save\_mode | Low memory mode. |
| p\_reserved | Reserved parameter. |

### 3.1.2 TY\_INIT\_PARAMS\_S

typedef struct

{

BYTE\_T encrypt\_key[ENCRYPT\_KEY\_LEN];

}TY\_INIT\_PARAMS\_S;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| encrypt\_key | Storage key |

### 3.1.3 GW\_ATTACH\_ATTR\_T

typedef struct

{

GW\_PERMIT\_DEV\_TP\_T tp;

CHAR\_T ver[SW\_VER\_LEN+1];

}GW\_ATTACH\_ATTR\_T;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| Tp | See definition of GW\_PERMIT\_DEV\_TP\_T. |
| ver | Software version. |

### 3.1.4 DEV\_DESC\_IF\_S

typedef struct

{

CHAR\_T id[DEV\_ID\_LEN+1];

CHAR\_Tsw\_ver[SW\_VER\_LEN+1];

CHAR\_T schema\_id[SCHEMA\_ID\_LEN+1];

CHAR\_T product\_key[PRODUCT\_KEY\_LEN+1];

USER\_DEV\_DTL\_DEF\_T uddd;

DEV\_TYPE\_T tp;

BOOL\_Tbind;

BOOL\_T sync;

BYTE\_T attr\_num;

GW\_ATTACH\_ATTR\_T attr[GW\_ATTACH\_ATTR\_LMT];

}DEV\_DESC\_IF\_S;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| id | Device ID |
| sw\_ver | Software version |
| schema\_id |  |
| product\_key | Device key |
| uddd | User-defined type |
| tp |  |
| bind |  |
| sync |  |
| attr\_num |  |
| attr | See section 3.1.3 "GW\_ATTACH\_ATTR\_T." |

### 3.1.5 DP\_PROP\_VAL\_S

typedef struct

{

INT\_T min;

INT\_T max;

SHORT\_T step;

USHORT\_T scale;

INT\_T value;

}DP\_PROP\_VAL\_S;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| min |  |
| max |  |
| step |  |
| scale |  |
| value |  |

### 3.1.6 DP\_PROP\_ENUM\_S

typedef struct

{

INT\_T cnt;

CHAR\_T \*\*pp\_enum;

INT\_T value;

}DP\_PROP\_ENUM\_S;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| cnt |  |
| pp\_enum |  |
| value | Enum |

### 3.1.7 DP\_PROP\_STR\_S

typedef struct

{

INT\_Tmax\_len;

CHAR\_T \*value;

}DP\_PROP\_STR\_S;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| max\_len | Length |
| value | String |

### 3.1.8 DP\_BOOL\_S

typedef struct

{

BOOL\_T value;

}DP\_BOOL\_S;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| value | Boolean |

### 3.1.9 DP\_PROP\_BITMAP

typedef struct

{

UINT\_T max\_len;

UINT\_Tvalue;

}DP\_PROP\_BITMAP;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| max\_len | Length |
| value | Bitmap |

### 3.1.10 TY\_RECV\_OBJ\_DP\_S

typedef struct {

DP\_CMD\_TYPE\_E cmd\_tp;

DP\_TRANS\_TYPE\_Tdtt\_tp;

CHAR\_T \*cid;

UINT\_T dps\_cnt;

TY\_OBJ\_DP\_S dps[0];

}TY\_RECV\_OBJ\_DP\_S;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| cmd\_tp | #define DP\_CMD\_LAN 0 // cmd from LAN；  #define DP\_CMD\_MQ 1 // cmd from MQTT；  #define DP\_CMD\_TIMER 2 // cmd from Local Timer；  #define DP\_CMD\_SCENE\_LINKAGE 3 // cmd from scene linkage； |
| dtt\_tp | #define DTT\_SCT\_UNC 0 // unicast；  #define DTT\_SCT\_BNC 1 // boardcast；  #define DTT\_SCT\_MNC 2 // multicast；  #define DTT\_SCT\_SCENE 3 // scene |
| cid | if(NULL == cid) then then the cid represents gwid |
| dps\_cnt |  |
| dps[0] | See section 3.1.11 "TY\_OBJ\_DP\_S." |

### 3.1.11 TY\_OBJ\_DP\_S

typedef struct {

BYTE\_T dpid;

DP\_PROP\_TP\_E type;

TY\_OBJ\_DP\_VALUE\_U value;

UINT\_T time\_stamp;

}TY\_OBJ\_DP\_S;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| dpid | DP ID |
| type | DP type  #define PROP\_BOOL 0  #define PROP\_VALUE 1  #define PROP\_STR 2  #define PROP\_ENUM 3  #define PROP\_BITMAP 4 |
| value | DP value. See section 3.5.1 "TY\_OBJ\_DP\_VALUE\_U." |
| time\_stamp | DP happen time. if 0, mean now |

### 3.1.12 TY\_DP\_QUERY\_S

typedef struct {

CHAR\_T \*cid;

UINT\_T cnt;

BYTE\_T dpid[0];

}TY\_DP\_QUERY\_S;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| cid | if(NULL == cid) then then the cid represents gwid |
| cnt | DP ID cnt if(0 == cnt) then query all object DP |
| dpid[0] | DP ID |

### 3.1.13 FW\_UG\_S

typedef struct {

DEV\_TYPE\_Ttp;

CHAR\_T fw\_url[FW\_URL\_LEN+1];

CHAR\_T fw\_md5[FW\_MD5\_LEN+1];

CHAR\_T sw\_ver[SW\_VER\_LEN+1];

UINT\_T file\_size;

}FW\_UG\_S;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| tp | Firmware type:  #define DEV\_NM\_ATH\_SNGL 0 // netlink module attach single device,(devid == gwid)；  #define DEV\_NM\_NOT\_ATH\_SNGL 1 // netlink module not attach single device,(devid == gwid)；  #define DEV\_ZB\_SNGL 2 // zigbee single device；  #define DEV\_RF433\_SNGL 3 // rf433 single device；  #define DEV\_BLE\_SNGL 4 // ble single device； |
| fw\_url | Firmware downloading address |
| fw\_md5 | Firmware encryption type |
| sw\_ver | Firmware version |
| file\_size | Firmware size |

### 3.2.1 IPC\_MEDIA\_INFO\_S

typedef struct

{

BOOL\_Tchannel\_enable[E\_CHANNEL\_MAX];

UINT\_T video\_fps[E\_CHANNEL\_VIDEO\_MAX];

UINT\_T video\_gop[E\_CHANNEL\_VIDEO\_MAX];

TUYA\_VIDEO\_BITRATE\_E video\_bitrate[E\_CHANNEL\_VIDEO\_MAX];

UINT\_T video\_width[E\_CHANNEL\_VIDEO\_MAX];

UINT\_T video\_height[E\_CHANNEL\_VIDEO\_MAX];

UINT\_T video\_freq[E\_CHANNEL\_VIDEO\_MAX];

TUYA\_CODEC\_ID video\_codec[E\_CHANNEL\_VIDEO\_MAX];

TUYA\_CODEC\_ID audio\_codec[E\_CHANNEL\_MAX];

UINT\_T audio\_fps[E\_CHANNEL\_MAX];

TUYA\_AUDIO\_SAMPLE\_E audio\_sample[E\_CHANNEL\_MAX];

TUYA\_AUDIO\_DATABITS\_E audio\_databits[E\_CHANNEL\_MAX];

TUYA\_AUDIO\_CHANNEL\_E audio\_channel[E\_CHANNEL\_MAX];

}IPC\_MEDIA\_INFO\_S;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| channel\_enable | Enable. |
| video\_fps | Frame rate. The default value is 30. |
| video\_gop | I-Frame interval. The default value is 60. |
| video\_bitrate | See definition of TUYA\_VIDEO\_BITRATE\_E. |
| video\_width | Width range, which is associated with the video coding formats SD and HD. The SD resolution is below 1280 x 720 pixels, and the HD resolution is 1280 x 720 pixels to 1920 x 1080 pixels. |
| video\_height | Height range, which is associated with the video coding formats SD and HD. The SD resolution is below 1280 x 720 pixels, and the HD resolution is 1280 x 720 pixels to 1920 x 1080 pixels. |
| video\_freq | Clock frequency. The default value is 90000. |
| video\_codec | See definition of TUYA\_CODEC\_ID. |
| audio\_codec | See definition of TUYA\_CODEC\_ID. |
| audio\_fps | Audio frame rate range. The default value is 25. |
| audio\_sample | See definition of TUYA\_AUDIO\_SAMPLE\_E. |
| audio\_databits | See definition of TUYA\_AUDIO\_DATABITS\_E. |
| audio\_channel | See definition of TUYA\_AUDIO\_CHANNEL\_E. |

### 3.2.2 AES\_HW\_CBC\_FUNC

typedef struct

{

Tuya\_CBC\_AES128\_Init init;

Tuya\_CBC\_AES128\_Encrypt encrypt;

Tuya\_CBC\_AES128\_Destroy destory;

}AES\_HW\_CBC\_FUNC;

**Note:**

TypedefINT\_T (**\*Tuya\_CBC\_AES128\_Init**)(VOID);

typedef INT\_T (\***Tuya\_CBC\_AES128\_Encrypt**)

(

IN BYTE\_T \*pdata\_in, //Data before encryption, which cannot be modified.

IN UINT\_T data\_len, //Data size before encryption

IN BYTE\_T \*pdata\_out, //Data destination address after encryption. A space has been applied for in the SDK and additional memory allocation is not required.

OUT UINT\_T \*pdata\_out\_len, //Data size after encryption

IN BYTE\_T \*pkey, //AES encryption key

IN BYTE\_T \*piv //AES CBC encryption IV

);

typedef INT\_T (\***Tuya\_CBC\_AES128\_Destroy**)(VOID);

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| init | See definition of Tuya\_CBC\_AES128\_Init. |
| encrypt | See definition of Tuya\_CBC\_AES128\_Encrypt. |
| destory | See definition of Tuya\_CBC\_AES128\_Destroy. |

### 3.3.1 SS\_QUERY\_DAY\_TS\_ARR\_S

typedef struct

{

UINT\_T file\_count;

SS\_FILE\_TIME\_TS\_S file\_arr[0];

} SS\_QUERY\_DAY\_TS\_ARR\_S;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| file\_count | Number of playback files on the current day |
| file\_arr | Playback file array |

### 3.3.2 SS\_FILE\_TIME\_TS\_S

typedef struct

{

UINT\_T start\_timestamp;

UINT\_T end\_timestamp;

} SS\_FILE\_TIME\_TS\_S;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| start\_timestamp | Playback file start timestamp (in seconds) |
| end\_timestamp | Playback file end timestamp (in seconds) |

typedef struct

{

MEDIA\_FRAME\_TYPE\_E type;

BYTE\_T \*p\_buf;

UINT\_T size;

UINT64\_T pts;

UINT64\_T timestamp;

}MEDIA\_FRAME\_S;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| type | Frame type. See definition of MEDIA\_FRAME\_TYPE\_E. |
| p\_buf | Callback data. |
| size | Data size. |
| pts | Displayed timestamp. |
| timestamp | Timestamp. |

### 3.4.1 Ring\_Buffer\_Node\_S

typedef struct

{

UINT\_T index;

MEDIA\_FRAME\_TYPE\_E type;

UCHAR\_T \*rawData;

UINT\_T size;

UINT64\_T pts;

UINT64\_T timestamp;

UINT\_T seqNo;

UCHAR\_T \*extraData;

UINT\_T extraSize;

}Ring\_Buffer\_Node\_S;

| **Parameter** | **Description** |
| --- | --- |
| index | Index |
| type | Frame type. See definition of MEDIA\_FRAME\_TYPE\_E. |
| rawData | Original data, such as audio and video data. |
| size | Data size. |
| pts | Displayed timestamp. |
| timestamp | Timestamp. The value is 1000 times the actual timestamp. The actual timestamp is this parameter value divided by 1000. |
| seqNo | Frame number. |
| \*extraData | Alternative parameter, which is meaningless. |
| extraSize | Alternative parameter, which is meaningless. |

### 3.4.2 S\_PRESET\_PTZ

typedef struct {

INT\_T pan;

INT\_T tilt;

INT\_T zoom;

}S\_PRESET\_PTZ;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| pan | Horizontal PTZ. The driver layer parameter value range is not limited and can be defined as needed. |
| tilt | Vertical PTZ. The driver layer parameter value range is not limited and can be defined as needed. |
| zoom | Zoom function. The driver layer parameter value range is not limited and can be defined as needed. |

### 3.4.3 S\_PRESET\_POSITION

typedef struct {

CHAR\_T id;

CHAR\_T name[32];

INT\_T mpId;

S\_PRESET\_PTZ ptz;

} S\_PRESET\_POSITION;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| id | Identifier for each preset point on the server. |
| tilt | Preset name. The value is a maximum of 31 characters, including Chinese characters. |
| zoom | Preset point index ID, which must be unique. |
| Ptz | PTZ position information. |

## 3.5 Joint Type Definition

### 3.5.1 TY\_OBJ\_DP\_VALUE\_U

typedef union {

INT\_T dp\_value; // valid when dp type is value

UINT\_Tdp\_enum; // valid when dp type is enum

CHAR\_T \*dp\_str; // valid when dp type is str

BOOL\_T dp\_bool; // valid when dp type is bool

UINT\_T dp\_bitmap; // valid when dp type is bitmap

}TY\_OBJ\_DP\_VALUE\_U;

|  |  |
| --- | --- |
| **Parameter** | **Description** |
| dp\_value | Valid when the DP is an integer |
| dp\_enum | Valid when the DP is an enum |
| dp\_str | Valid when the DP is a string |
| dp\_bool | Valid when the DP is Boolean |
| dp\_bitmap | Valid when the DP is a bitmap |

# 4 Interface Definitions

4.1 General Interface Definitions

### 4.1.1 tuya\_ipc\_get\_sdk\_info

**Function prototype**

CHAR\_T \*tuya\_ipc\_get\_sdk\_info(

VOID

);

**Function description**

This interface obtains the Tuya SDK compilation information.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| Info | Tuya SDK compilation information, SDK version, compiling time, and supported functions |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.1.2 tuya\_ipc\_init\_sdk

**Function prototype**

OPERATE\_RET tuya\_ipc\_init\_sdk(

IN CONST TUYA\_IPC\_ENV\_VAR\_S \*p\_var

);

**Function description**

This interface verifies and initializes the required Tuya SDK operating environment.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| p\_var | Running environment configuration structure pointer | Input parameter | Yes | See definition of TUYA\_IPC\_ENV\_VAR\_S. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.1.3 tuya\_ipc\_start\_sdk

**Function prototype**

OPERATE\_RET tuya\_ipc\_start\_sdk(

IN CONST WIFI\_INIT\_MODE\_E wifi\_mode,

IN CONST char \*pToken

);

**Function description**

This interface starts the Tuya SDK in WIFI\_DEVICE mode.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| wifi\_mode | Device network configuration mode | Input parameter | Yes | See definition of WIFI\_INIT\_MODE\_E. |
| pToken | Token | Input parameter | Yes |  |

**Returned value**

|  |  |
| --- | --- |
| **Returned value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.1.4 tuya\_ipc\_upgrade\_sdk

**Function prototype**

OPERATE\_RET tuya\_ipc\_upgrade\_sdk(

IN CONST FW\_UG\_S \*fw,

IN CONST GET\_FILE\_DATA\_CB get\_file\_cb,

IN CONST UPGRADE\_NOTIFY\_CBupgrd\_nofity\_cb,

IN PVOID\_T pri\_data

);

**Function description**

This interface upgrades the local firmware.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| fw | Firmware information | Input parameter | Yes | See section 3.1.13 "FW\_UG\_S." |
| get\_file\_cb | Callback function for firmware file download | Input parameter | Yes | See section 4.1.12 "GET\_FILE\_DATA\_CB." |
| upgrd\_nofity\_cb | Callback function for notifications on the completion of downloading firmware | Input parameter | Yes | See section 4.1.13 "UPGRADE\_NOTIFY\_CB." |
| pri\_data | Additional parameter when the callback function is called. | Input parameter | Yes |  |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.1.5 tuya\_ipc\_get\_register\_status

**Function prototype**

IPC\_REGISTER\_STATUS tuya\_ipc\_get\_register\_status(

VOID

);

**Function description**

This interface obtains IPC registration and activation states.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned value** | **Description** |
| IPC\_REGISTER\_STATUS | See IPC\_REGISTER\_STATUS. |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.1.6 DEV\_OBJ\_DP\_CMD\_CB

**Function prototype**

typedef VOID (\*DEV\_OBJ\_DP\_CMD\_CB)(

IN CONST TY\_RECV\_OBJ\_DP\_S \*dp

);

**Function description**

This is a Tuya command distribution function.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| dp | Original DP information | Input parameter | Yes | See definition of TY\_RECV\_OBJ\_DP\_S. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.1.7 DEV\_DP\_QUERY\_CB

**Function prototype**

Typedef VOID (\*DEV\_DP\_QUERY\_CB)(

IN CONST TY\_DP\_QUERY\_S\*dp\_qry

);

**Function description**

This is a Tuya query distribution function.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| dp\_qry | Querying information | Input parameter | Yes | See section 3.1.12 "TY\_DP\_QUERY\_S." |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.1.8 GW\_STATUS\_CHANGED\_CB

**Function prototype**

typedef VOID (\*GW\_STATUS\_CHANGED\_CB)(

IN CONST GW\_STATUS\_E status

);

**Function description**

This is a device status change callback function. The value can be **NULL.**

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| dp\_qry | Gateway status | Input parameter | Yes | #define GW\_RESET 0 // gw reset；  #define GW\_ACTIVED 1 // gw actived；  #define GW\_FIRST\_START 2 // start tuya-sdk in the first time.；  #define GW\_NORMAL 3 // tuya-sdk is activated and started |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.1.9 GW\_UG\_INFORM\_CB

**Function prototype**

typedef VOID (\*GW\_UG\_INFORM\_CB)(

IN CONST FW\_UG\_S \*fw

);

**Function description**

This is a callback function for getting the firmware upgrade information in IPC mode. The value can be **NULL.**

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| fw | Firmware information | Input parameter | Yes | See section 3.1.13 "FW\_UG\_S." |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.1.10 TUYA\_RST\_INFORM\_CB

**Function prototype**

Typedef VOID (\*TUYA\_RST\_INFORM\_CB)(

GW\_RESET\_TYPE\_E from

);

**Function description**

This is a callback function for resetting the IPC in IPC mode. The value can be **NULL.**

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| from | Firmware information | Input parameter | Yes | See section 3.1.13 "FW\_UG\_S." |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.1.11 TUYA\_RESTART\_INFORM\_CB

**Function prototype**

typedef VOID (\*TUYA\_RESTART\_INFORM\_CB)(

VOID

);

**Function description**

This is a callback function for restarting the IPC progress in IPC mode. The value can be **NULL.**

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.1.12 GET\_FILE\_DATA\_CB

**Function prototype**

typedef OPERATE\_RET (\*GET\_FILE\_DATA\_CB)(

IN CONST FW\_UG\_S \*fw,

IN CONST UINT\_T total\_len,

IN CONST UINT\_T offset,

IN CONST BYTE\_T \*data,

IN CONST UINT\_T len,

OUT UINT\_T \*remain\_len,

IN PVOID\_T pri\_data

);

**Function description**

This is a callback function for downloading firmware files.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| fw | Firmware information | Input parameter | Yes | See section 3.1.13 "FW\_UG\_S." |
| total\_len | Firmware size | Input parameter | Yes |  |
| offset | Downloaded packet offset | Input parameter | Yes |  |
| data | Packet data | Input parameter | Yes |  |
| len | Packet length | Input parameter | Yes |  |
| remain\_len | Volume of data to download | Output parameter | Yes |  |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.1.13 UPGRADE\_NOTIFY\_CB

**Function prototype**

typedef VOID (\*UPGRADE\_NOTIFY\_CB)(

IN CONST FW\_UG\_S \*fw,

IN CONST INT\_T download\_result,

IN PVOID\_T pri\_data

);

**Function description**

This is a callback function for notifications on the completion of downloading firmware.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| fw | Firmware information | Input parameter | Yes | See section 3.1.13 "FW\_UG\_S." |
| download\_result | Downloading result | Input parameter | Yes | The value 0 indicates that the firmware has been downloaded and a non-zero value indicates that the download has failed. |
| pri\_data |  | Input parameter | Yes |  |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

## 4.2 Time Interface Definitions

### 4.2.1 tuya\_ipc\_get\_service\_time

**Function prototype**

OPERATE\_RET tuya\_ipc\_get\_service\_time(

OUT UINT\_T \*time\_utc,

OUT INT\_T \*time\_zone

);

**Function description**

This interface obtains the server time that is the same as the time in the app.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| time\_utc | UTC | Output parameter | Yes | None |
| time\_zone | Time zone, which is returned in seconds. | Output parameter | Yes | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.2.2 tuya\_ipc\_check\_in\_dls

**Function prototype**

OPERATE\_RET tuya\_ipc\_check\_in\_dls(

IN UINT\_T time\_utc,

OUT BOOL\_T \* pIsDls

);

**Function description**

This interface verifies whether it is in daylight saving time.

Note: The function is available in the US.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| time\_utc | UTC | Input parameter | Yes | UTC |
| pIsDls | Indicates whether it is in daylight saving time  **TRUE**: Yes **FALSE**: No | Output parameter | Yes | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.2.3 tuya\_ipc\_get\_local\_time

**Function prototype**

OPERATE\_RET tuya\_ipc\_get\_local\_time(

IN UINT\_T inTime,

OUT struct tm \*localTime

);

**Function description**

This interface converts time to local time in UTC.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| inTime | UTC, which is returned in seconds. | Input parameter | Yes | UTC, in seconds |
| localTime | Local time in UTC, in the time structure format. The output contains the year, month, date, day, hour, minute, second, and week. | Output parameter | Yes | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.2.4 tuya\_ipc\_get\_utc\_tm

**Function prototype**

OPERATE\_RET tuya\_ipc\_get\_utc\_tm(

OUT struct tm \*localTime

);

**Function description**

This interface obtains the UTC.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| localTime | UTC, which is returned in the year-month-day format. | Output parameter | Yes | UTC after 1970. UTC before 1970 is not supported. If the parameter is set to UTC before 1970, an error will occur. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.2.5 tuya\_ipc\_get\_utc\_time

**Function prototype**

OPERATE\_RET tuya\_ipc\_get\_utc\_time(

OUT UINT\_T \*time\_utc

);

**Function description**

This interface obtains the UTC.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| time\_utc | UTC, which is returned in seconds. | Output parameter | Yes | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

## 4.3 Message Interface Definitions

### 4.3.1 tuya\_ipc\_dp\_report

**Function prototype**

OPERATE\_RET tuya\_ipc\_dp\_report(

IN CONST CHAR\_T \*dev\_id,

IN BYTE\_T dp\_id,

IN DP\_PROP\_TP\_E type,

IN VOID \* pVal,

IN CONST UINT\_Tcnt

);

**Function description**

This interface reports Tuya messages.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| dev\_id | device id | Input parameter | No | The value can be **NULL**. |
| dp\_id | DP ID | Input parameter | Yes | DP ID, for example, 134 (motion detection) |
| type | Reporting type | Input parameter | Yes | #define PROP\_BOOL 0;  #define PROP\_VALUE 1;  #define PROP\_STR 2  #define PROP\_ENUM 3;  #define PROP\_BITMAP 4; |
| pVal | Reported value address | Input parameter | Yes |  |
| cnt | Number of reporting times | Input parameter | Yes |  |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.3.2 tuya\_ipc\_dp\_report\_sync

**Function prototype**

OPERATE\_RET tuya\_ipc\_dp\_report\_sync(

IN CONST CHAR\_T \*p\_json\_data,

IN CONST BOOL\_T force,

IN CONST UINT\_T timeout

);

**Function description**

This interface reports Tuya messages.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| p\_json\_data | Message content pointer | Input parameter | Yes | JSON format |
| force | Indicates whether a message is forcibly sent | Input parameter | Yes | **0**: No **1**: Yes |
| timeout | Timeout | Input parameter | Yes | Timeout range. 5s is recommended. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.3.3 tuya\_ipc\_notification\_content\_upload\_from\_buffer

**Function prototype**

OPERATE\_RET tuya\_ipc\_notification\_content\_upload\_from\_buffer(

IN NOTIFICATION\_CONTENT\_TYPE\_E type,

IN CHAR\_T \*data,

IN INT\_T data\_len,

OUT VOID\*message

);

**Function description**

This interface pushes buffer audio and video data to Tuya Cloud.

Note:

1. It is a blocking function that can be used securely in multiple threads.

2. From SDK 4.3.0, this interface can no longer be used to upload video files.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| type | Image or video type | Input parameter | Yes | See definition of NOTIFICATION\_CONTENT\_TYPE\_E. |
| data | Data pointer | Input parameter | Yes | Image data path |
| data\_len | Data length | Input parameter | Yes | Image data size: ≤ 150 KB |
| message | Message storage pointer for notification information | Output parameter | Yes | Message returned by the server |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.3.4 tuya\_ipc\_notification\_content\_upload\_from\_file

**Function prototype**

OPERATE\_RET tuya\_ipc\_notification\_content\_upload\_from\_file(

IN CHAR\_T \*name,

IN NOTIFICATION\_CONTENT\_TYPE\_E type,

OUT VOID \*message

);

**Function description**

This interface uploads the content body of a notification message to Tuya Cloud.

Note:

1. It is a blocking function that can be used securely in multiple threads.

2. This interface cannot be used to upload videos.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| name | Image path and name | Input parameter | Yes | **Not limited** |
| type | Image type | Input parameter | Yes | See definition of NOTIFICATION\_CONTENT\_TYPE\_E.  The value can be 1 or 2, but not 0. |
| message | Message storage pointer for notification information | Output parameter | Yes | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.3.5 tuya\_ipc\_notification\_message\_upload

**Function prototype**

OPERATE\_RET tuya\_ipc\_notification\_message\_upload(

IN CONST BYTE\_T notification,

IN VOID\*message,

IN UINT\_T time\_out

);

**Function description**

This interface automatically generates messages and pushes them to the app based on the push type and content.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| notification | Push type | Input parameter | Yes | In SDK 4.3.0 and earlier versions, this interface can only be used to send IPC DP 115 (motion detection), and not doorbell DP 115 or other message types. |
| message | Message pointer | Input parameter | Yes | Message returned by the server |
| time\_out | Timeout | Input parameter | Yes | 5s is recommended. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.3.6 tuya\_ipc\_snapshot\_message\_upload

**Function prototype**

OPERATE\_RET tuya\_ipc\_snapshot\_message\_upload(

IN CONST BYTE\_T snapshot,

IN VOID \*message,

IN UINT\_T time\_out

);

**Function description**

This interface pushes the screenshot information to the app when the doorbell is pressed.

Note: This function is only available for doorbells.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| snapshot | Push type | Input parameter | Yes | Doorbell push screenshot (DP 154) |
| message | Message pointer | Input parameter | Yes | Result returned by the server |
| time\_out | Timeout | Input parameter | Yes | The default value is 5, in seconds. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.3.7 tuya\_ipc\_report\_living\_msg

**Function prototype**

OPERATE\_RET tuya\_ipc\_report\_living\_msg(

IN UINT\_T error\_code,

IN UINT\_T force,

IN UINT\_T timeout

);

**Function description**

This interface notifies the app of the live video information.

Note: The interface is used for Minrui devices only. The internal encapsulation 305 protocol and **error\_code** are meaningless.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| error\_code | Error code | Input parameter | Yes |  |
| force | Indicates whether a message is forcibly pushed | Input parameter | Yes | **0**: No **1**: Yes |
| timeout | Timeout | Input parameter | Yes | Time range. 5s is recommended. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

## 4.4 Device Management Interface Definitions

### 4.4.1 tuya\_ipc\_get\_wakeup\_data

**Function prototype**

OPERATE\_RET tuya\_ipc\_get\_wakeup\_data(

INOUT BYTE\_T \*wakeup\_data\_arr,

INOUT UINT\_T \*p\_len

);

**Function description**

This interface obtains a device's sleep mode data.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| wakeup\_data\_arr | Wakeup data pushed to the device in sleep mode | Output parameter | Yes |  |
| p\_len | Length of the wakeup data pushed to the device in sleep mode | Output parameter | Yes |  |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.4.2 tuya\_ipc\_get\_heartbeat\_data

**Function prototype**

OPERATE\_RET tuya\_ipc\_get\_heartbeat\_data(

INOUT BYTE\_T \*heartbeat\_data\_arr,

INOUT UINT\_T \*p\_len

);

**Function description**

This interface obtains the heartbeat data of a device in sleep mode.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| heartbeat\_data\_arr | Heartbeat data for a device in sleep mode | Input parameter | Yes |  |
| p\_len | Length of the heartbeat data for a device in sleep mode | Input parameter | Yes |  |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.4.3 tuya\_ipc\_direct\_connect

**Function prototype**

OPERATE\_RET tuya\_ipc\_direct\_connect(

IN CONST CHAR\_T\*p\_str,

IN CONST TUYA\_IPC\_DIRECT\_CONNECT\_TYPE\_E source

);

**Function description**

This interface is used for direct network access, for example, QR code net-pairing and sound wave net-pairing.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| p\_str | Network configuration parameter | Input parameter | Yes | Format: {"p":"20112012pw1","s":"GAOKE\_57093C","t":"AYkbx3mD298GJi"} **p**: password **s**: SSID **t**: token |
| source | Network configuration mode | Input parameter | Yes | Currently, only QR code net-pairing is supported. For details, see definition of TUYA\_IPC\_DIRECT\_CONNECT\_QRCODE. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.4.4 tuya\_ipc\_set\_net\_info\_nofify\_cb

**Function prototype**

OPERATE\_RET tuya\_ipc\_set\_net\_info\_nofify\_cb(

VOID

);

**Function description**

This interface sets a callback function for notification of the network configuration information.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned Value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.4.5 tuya\_ipc\_get\_mqtt\_status

**Function prototype**

OPERATE\_RET tuya\_ipc\_get\_mqtt\_status(

VOID

);

**Function description**

This interface obtains the MQTT connection status.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| MQTT status | **0**: offline **1**: online |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.4.6 tuya\_ipc\_book\_wakeup\_topic

**Function prototype**

OPERATE\_RET tuya\_ipc\_book\_wakeup\_topic(

VOID

);

**Function description**

This interface is used for subscribed wakeup events.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.4.7 tuya\_ipc\_get\_mqtt\_socket\_fd

**Function prototype**

INT\_T tuya\_ipc\_get\_mqtt\_socket\_fd(

VOID

);

**Function description**

This interface obtains the MQTT socket FD.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| Socket\_fd | MQTT socket FD |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.4.8 tuya\_ipc\_set\_log\_attr

**Function prototype**

OPERATE\_RET tuya\_ipc\_set\_log\_attr(

IN CONST INT\_T log\_level,

CHAR\_T \*filename

);

**Function description**

This interface sets the log printing level.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| log\_level | Log printing level | Input parameter | Yes | The value range is 0 to 5. A numerically higher value indicates more log content.  Level 0: error message, which is unexpected when the program is running properly Level 1: warning Level 2: information that needs attention Level 3: notification Level 4: program running debugging information, which is deleted in the release version Level 5: program running path information, which is deleted in the release version Logs at levels 1, 2, and 3 are not used. |
| filename | File path | Input parameter | No | This parameter is temporarily meaningless. The saving log function is available. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.4.9 tuya\_ipc\_get\_free\_ram

**Function prototype**

OPERATE\_RET tuya\_ipc\_get\_free\_ram(

IN ULONG\_T \*free

);

**Function description**

This interface queries available system memory.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| free | Available memory, in kilobytes (KB) | Output parameter | Yes | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_com\_defs.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

## 4.5 Cloud Storage Interface Definitions

### 4.5.1 tuya\_ipc\_cloud\_storage\_init

**Function prototype**

OPERATE\_RET tuya\_ipc\_cloud\_storage\_init(

IN IPC\_MEDIA\_INFO\_S\*media\_setting,

IN AES\_HW\_CBC\_FUNC \*aes\_func

);

**Function description**

This interface initializes the cloud storage function.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| media\_setting | Multimedia stream configuration information. Specify the maximum value if it uses a variable bit rate or is modifiable during operation. | Input parameter | Yes | See section 3.2.1 "IPC\_MEDIA\_INFO\_S." |
| aes\_func | Encryption information | Input parameter | Yes | See section 3.2.2 "AES\_HW\_CBC\_FUNC." |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.5.2 tuya\_ipc\_cloud\_storage\_uninit

**Function prototype**

VOID tuya\_ipc\_cloud\_storage\_uninit(

VOID

);

**Function description**

This interface deinitializes the cloud storage function.

Note: You are advised to use the interface to release resources before the device restarts.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.5.3 tuya\_ipc\_cloud\_storage\_get\_store\_mode

**Function prototype**

ClOUD\_STORAGE\_TYPE\_E tuya\_ipc\_cloud\_storage\_get\_store\_mode(

VOID

);

**Function description**

This interface obtains the current cloud storage order type.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| Cloud storage order type | See definition of ClOUD\_STORAGE\_TYPE\_E. |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.5.4 tuya\_ipc\_cloud\_storage\_event\_start

**Function prototype**

OPERATE\_RET tuya\_ipc\_cloud\_storage\_event\_start(

IN CHAR\_T \*snapshot\_buffer,

IN UINT\_T snapshot\_size,

IN ClOUD\_STORAGE\_EVENT\_TYPE\_E type

);

**Function description**

This interface reports the start of an event.

Note: You need to enable the cloud storage service; otherwise, an invalid event error (error code –2000) will be reported.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| snapshot\_buffer | Snapshot when the current event occurs | Input parameter | Yes | Image data cache pointer |
| snapshot\_size | Size of the snapshot when the current event occurs | Input parameter | Yes | ≤ 100 KB |
| type | Event type | Input parameter | Yes | See definition of ClOUD\_STORAGE\_EVENT\_TYPE\_E. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.5.5 tuya\_ipc\_cloud\_storage\_event\_stop

**Function prototype**

OPERATE\_RET tuya\_ipc\_cloud\_storage\_event\_stop(

VOID

);

**Function description**

This interface reports the end of the current event. Cross reporting of the start and end of multiple events is not supported.

Note: In SDK 4.3.2, all on-going events can be ended simultaneously.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.5.6 tuya\_ipc\_cloud\_storage\_get\_event\_status

**Function prototype**

EVENT\_STATUS\_E tuya\_ipc\_cloud\_storage\_get\_event\_status(

ClOUD\_STORAGE\_EVENT\_TYPE\_E type

);

**Function description**

This interface reports the current event status.

Note: In SDK 4.3.2, the input parameter **type** has been added. Querying an event requires the event ID.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| type | Event type | Input parameter | Yes | See definition of EVENT\_STATUS\_E. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| Event status | See definition of EVENT\_STATUS\_E. |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.5.7 tuya\_ipc\_cloud\_storage\_pause

**Function prototype**

VOID tuya\_ipc\_cloud\_storage\_pause(

VOID

);

**Function description**

This interface pauses cloud storage.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.5.8 tuya\_ipc\_cloud\_storage\_resume

**Function prototype**

VOID tuya\_ipc\_cloud\_storage\_resume(

VOID

);

**Function description**

This interface resumes cloud storage. It is used with tuya\_ipc\_cloud\_storage\_pause.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.5.9 tuya\_ipc\_cloud\_storage\_event\_add

**Function prototype**

EVENT\_ID tuya\_ipc\_cloud\_storage\_event\_add(

CHAR\_T \* snapshot\_buffer,

UINT\_T snapshot\_size,

ClOUD\_STORAGE\_EVENT\_TYPE\_E type,

UINT\_T max\_duration

);

**Function description**

This interface adds cloud storage events.

Note: This interface is new in SDK 4.3.2. It can be used to add a maximum of six cloud storage events and is recommended. The tuya\_ipc\_cloud\_storage\_event\_start interface can be used to add a single event.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| snapshot\_buffer | Snapshot when the current event occurs | Input parameter | Yes | Buffer pointer |
| snapshot\_size | Image size | Input parameter | Yes | 1 KB to 100 KB |
| type | Event type | Input parameter | Yes | See definition of ClOUD\_STORAGE\_EVENT\_TYPE\_E. |
| max\_duration | Maximum event duration | Input parameter | Yes | 0s to 300s |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| EVENT\_ID | Event ID, ranging from 0 to 5. If the event creation fails, INVALID\_EVENT\_ID 0xFFFF is returned. |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_cloud\_storage.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.5.10 tuya\_ipc\_cloud\_storage\_event\_delete

**Function prototype**

OPERATE\_RET tuya\_ipc\_cloud\_storage\_event\_delete(

EVENT\_ID event\_id

);

**Function description**

This interface stops a specified cloud storage event.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| event\_id | Event ID | Input parameter | Yes | Value: 0 to 5 |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_cloud\_storage.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

## 4.6 SD Card Storage Interface Definitions

### 4.6.1 tuya\_ipc\_ss\_init

**Function prototype**

OPERATE\_RET tuya\_ipc\_ss\_init(

IN CHAR\_T \*base\_path,

IN IPC\_MEDIA\_INFO\_S \*media\_setting,

IN UINT\_T max\_event\_per\_day

);

**Function description**

This interface initializes the SD card storage logic.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| base\_path | Root directory of the SD card | Input parameter | Yes |  |
| media\_setting | Audio and video configuration information | Input parameter | Yes | See section 3.2.1 "IPC\_MEDIA\_INFO\_S." |
| max\_event\_per\_day | Maximum number of events on a single day | Input parameter | Yes | If the number of events on a single day exceeds the value, playback will fail due to storage exceptions. If this parameter is set to a large value, it will take a long time to respond to the app, causing user experience to deteriorate. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.2 tuya\_ipc\_ss\_uninit

**Function prototype**

VOID tuya\_ipc\_ss\_uninit(

VOID

);

**Function description**

This interface deinitializes the local storage to release memory resources.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.3 tuya\_ipc\_ss\_set\_write\_mode

**Function prototype**

OPERATE\_RET tuya\_ipc\_ss\_set\_write\_mode(

IN CONST STREAM\_STORAGE\_WRITE\_MODE\_Ewrite\_mode

);

**Function description**

This interface sets the local storage mode.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| write\_mode | Full-time recording or event-triggered recording | Input parameter | Yes | See definition of STREAM\_STORAGE\_WRITE\_MODE\_E. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.4 tuya\_ipc\_ss\_get\_write\_mode

**Function prototype**

STREAM\_STORAGE\_WRITE\_MODE\_E tuya\_ipc\_ss\_get\_write\_mode(

VOID

);

**Function description**

This interface obtains the current local storage mode.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| Storage mode | See definition of STREAM\_STORAGE\_WRITE\_MODE\_E. |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.5 tuya\_ipc\_ss\_trigger\_event

**Function prototype**

OPERATE\_RET tuya\_ipc\_ss\_trigger\_event(

UINT\_Tevent\_duration

);

**Function description**

This interface triggers a local SD card to record events with a fixed duration. This interface is not recommended and may be discarded in later versions.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| event\_duration | Duration | Input parameter | Yes |  |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.6 tuya\_ipc\_ss\_start\_event

**Function prototype**

OPERATE\_RET tuya\_ipc\_ss\_start\_event(

VOID

);

**Function description**

This interface starts the recording of the local SD card event. It must be used with tuya\_ipc\_ss\_stop\_event.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.7 tuya\_ipc\_ss\_stop\_event

**Function prototype**

OPERATE\_RET tuya\_ipc\_ss\_stop\_event (

VOID

);

**Function description**

This interface stops the recording of the local SD card event. It must be used with tuya\_ipc\_ss\_start\_event.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.8 tuya\_ipc\_ss\_delete\_oldest\_event

**Function prototype**

DELETE\_LEVEL\_E tuya\_ipc\_ss\_delete\_oldest\_event(

VOID

);

**Function description**

This interface deletes the oldest event file when there is insufficient disk space. In continuous storage, the video is divided into virtualized event segments with a fixed length.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| Deleted event information | See definition of DELETE\_LEVEL\_E. |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.9 tuya\_ipc\_ss\_delete\_all\_files

**Function prototype**

VOID tuya\_ipc\_ss\_delete\_all\_files(

VOID

);

**Function description**

This interface deletes files stored locally.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.10 tuya\_ipc\_ss\_get\_status

**Function prototype**

STORAGE\_STATUS\_E tuya\_ipc\_ss\_get\_status(

VOID

);

**Function description**

This interface obtains the storage running status.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| Storage status | See definition of STORAGE\_STATUS\_E. |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.11 tuya\_ipc\_pb\_query\_by\_month

**Function prototype**

OPERATE\_RET tuya\_ipc\_pb\_query\_by\_month(

IN USHORT\_T query\_year,

IN USHORT\_T query\_month,

OUT UINT\_T \*p\_return\_days

);

**Function description**

This interface queries the number of days for storing data locally on the SD card in a specified year and month.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| query\_year | Year | Input parameter | Yes | Year |
| query\_month | Month | Input parameter | Yes | Month |
| p\_return\_days | Effective number of storage days | Output parameter | Yes | Number of days |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.12 tuya\_ipc\_pb\_query\_by\_day

**Function prototype**

OPERATE\_RET tuya\_ipc\_pb\_query\_by\_day(

IN UINT\_T pbIdx,

IN USHORT\_T year,

IN USHORT\_T month,

IN UCHAR\_T day,

OUT SS\_QUERY\_DAY\_TS\_ARR\_S \*\*pTsArr

);

**Function description**

This interface checks details of data stored locally on the SD card on a specified date.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| pbIdx | Playback ID | Input parameter | Yes |  |
| year | Year | Input parameter | Yes | Year |
| month | Month | Input parameter | Yes | Month |
| day | Day | Input parameter | Yes | Day |
| pTsArr | Valid data details | Output parameter | Yes |  |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.13 tuya\_ipc\_pb\_query\_free\_ts\_arr

**Function prototype**

OPERATE\_RET tuya\_ipc\_pb\_query\_free\_ts\_arr(

IN SS\_QUERY\_DAY\_TS\_ARR\_S \*p\_query\_ts\_arr

);

**Function description**

This interface releases the memory returned by the SDK.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| p\_query\_ts\_arr | Memory to be released | Input parameter | Yes | See section 3.3.1 "SS\_QUERY\_DAY\_TS\_ARR\_S." |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.14 SS\_PB\_EVENT\_CB

**Function prototype**

typedef VOID (\*SS\_PB\_EVENT\_CB)(

IN UINT\_T pb\_idx,

IN SS\_PB\_EVENT\_E pb\_event,

IN PVOID\_T args

);

**Function description**

This interface defines the callback function for event playback.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| pb\_idx | Playback ID | Input parameter | Yes | Value: 0 to 65535 |
| pb\_event | Playback event ID | Input parameter | Yes | See definition of SS\_PB\_EVENT\_E. |
| args | Reserved parameter | Input parameter | Yes |  |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.15 SS\_PB\_GET\_MEDIA\_CB

**Function prototype**

typedef VOID (\*SS\_PB\_GET\_MEDIA\_CB)(

IN UINT\_T pb\_idx,

IN CONST MEDIA\_FRAME\_S \*p\_frame

);

**Function description**

This interface obtains playback data.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| pb\_idx | Playback ID | Input parameter | Yes | Value: 0 to 65535 |
| p\_frame  s | Playback data | Input parameter | Yes | See definition of MEDIA\_FRAME\_S |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.16 tuya\_ipc\_ss\_pb\_start

**Function prototype**

OPERATE\_RET tuya\_ipc\_ss\_pb\_start(

IN UINT\_T pb\_idx,

IN SS\_PB\_EVENT\_CB event\_cb,

IN SS\_PB\_GET\_MEDIA\_CB video\_cb,

IN SS\_PB\_GET\_MEDIA\_CBaudio\_cb

);

**Function description**

This interface starts a new playback.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| pb\_idx | Playback ID | Input parameter | Yes | Value: 0 to 65535 |
| event\_cb | Callback function for event playback | Input parameter | Yes | See definition of SS\_PB\_EVENT\_CB. |
| video\_cb | Callback function for video playback | Input parameter | Yes | See definition of IN SS\_PB\_GET\_MEDIA\_CB. |
| audio\_cb | Callback function for audio playback | Input parameter | Yes | See definition of IN SS\_PB\_GET\_MEDIA\_CB. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.17 tuya\_ipc\_ss\_pb\_set\_status

**Function prototype**

OPERATE\_RET tuya\_ipc\_ss\_pb\_set\_status(

IN UINT\_T pb\_idx,

IN SS\_PB\_STATUS\_E new\_status

);

**Function description**

This interface sets the playback attribute.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| pb\_idx | Playback ID | Input parameter | Yes | Value: 0 to 65535 |
| new\_status | Status | Input parameter | Yes | See definition of SS\_PB\_STATUS\_E. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.18 tuya\_ipc\_ss\_pb\_stop

**Function prototype**

OPERATE\_RET tuya\_ipc\_ss\_pb\_stop(

IN UINT\_T pb\_idx

);

**Function description**

This interface stops a playback.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| pb\_idx | Playback ID | Input parameter | Yes | Value: 0 to 65535 |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.19 tuya\_ipc\_ss\_pb\_seek

**Function prototype**

OPERATE\_RET tuya\_ipc\_ss\_pb\_seek(

IN UINT\_T pb\_idx,

IN SS\_FILE\_TIME\_TS\_S \*pb\_file\_info,

IN UINT\_T play\_timestamp

);

**Function description**

This interface seeks a specified playback.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| pb\_idx | Playback ID | Input parameter | Yes | Value: 0 to 65535 |
| pb\_file\_info | File to seek | Input parameter | Yes | See section 3.3.2 "SS\_FILE\_TIME\_TS\_S." |
| play\_timestamp | Absolute timestamp for lookup | Input parameter | Yes | Timestamp |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.6.20 tuya\_ipc\_ss\_pb\_stop\_all

**Function prototype**

OPERATE\_RET tuya\_ipc\_ss\_pb\_stop\_all(

VOID

);

**Function description**

This interface stops all playbacks.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| None | None | None | None | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

## 4.7 Audio and Video Interface Definitions

### 4.7.1 tuya\_ipc\_ring\_buffer\_init

**Function prototype**

OPERATE\_RET tuya\_ipc\_ring\_buffer\_init(

CHANNEL\_E channel,

UINT\_T bitrate,

UINT\_T fps,

UINT\_T max\_buffer\_seconds,

FUNC\_REQUEST\_I\_FRAME requestIframeCB

);

**Function description**

This interface initializes a ring buffer.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| channel | The ring buffer supports multiple channels that apply to IPC scenarios, such as main streams, substreams, and triple streams. | Input parameter | Yes | See definition of CHANNEL\_E. Currently, 16 channels are supported. The value 1 indicates the video main stream, and the value 9 indicates the audio main stream. Values of CHANNEL\_E are for reference only. Redefine the channels based on your requirements. |
| bitrate | Code rate of the current stream; unit: kbits/s | Input parameter | Yes | The recommended range is 64 kbits/s to 1536 kbits/s. Set the value based on the capability of your product. |
| fps | Frames per second | Input parameter | Yes | The value range is not limited, and the value 25 is recommended. Set the value based on your product requirements. |
| max\_buffer\_seconds | Maximum cache seconds, one I-Frame interval of up to 10s. When the value is set to **0**, the duration is 10 seconds by default. The default setting is recommended. | Input parameter | Yes |  |
| requestIframeCB | Callback function for applying for I-Frames, which accelerates the I-Frame plotting when a new user accesses the SDK and sends a request. If the value is **NULL**, the system will not automatically apply for I-Frames. | Input parameter | Yes | See section 4.7.13 "FUNC\_REQUEST\_I\_FRAME." |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.7.2 tuya\_ipc\_ring\_buffer\_append\_data

**Function prototype**

OPERATE\_RET tuya\_ipc\_ring\_buffer\_append\_data(

CHANNEL\_E channel,

UCHAR\_T \*addr,

UINT\_T size,

MEDIA\_FRAME\_TYPE\_E type,

UINT64\_T pts

);

**Function description**

This interface adds new data to the ring buffer.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| channel | The ring buffer supports multiple channels that apply to IPC scenarios, such as main streams, substreams, and triple streams. | Input parameter | Yes | See definition of CHANNEL\_E. |
| addr |  | Input parameter | Yes | Audio and video stream data |
| size | Uploaded data size | Input parameter | Yes | < 200 x 1024 pixels |
| type | Frame type | Input parameter | Yes | See definition of MEDIA\_FRAME\_TYPE\_E. |
| pts |  | Input parameter | Yes | Timestamp of synchronization between the device and server |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.7.3 tuya\_ipc\_ring\_buffer\_append\_extra\_data

**Function prototype**

OPERATE\_RET tuya\_ipc\_ring\_buffer\_append\_extra\_data(

CHANNEL\_E channel,

UCHAR\_T \*dataAddr,

UINT\_T dataSize

);

**Function description**

This interface adds additional data to the ring buffer.

Note: This interface stores the generated audio and video data of the same frame in different formats. For example, the same code is stored in both PCM and AAC formats.

The data is stored in the **extraData** and **extraSize** fields of the node.

Note: This interface must be used with tuya\_ipc\_ring\_buffer\_append\_data to ensure that the data is stored in the same frame node.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| channel | The ring buffer supports multiple channels that apply to IPC scenarios, such as main streams, substreams, and triple streams. | Input parameter | Yes | See definition of CHANNEL\_E. |
| dataAddr |  | Input parameter | Yes |  |
| dataSize | Uploaded data size | Input parameter | Yes |  |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.7.4 tuya\_ipc\_ring\_buffer\_get\_video\_frame

**Function prototype**

Ring\_Buffer\_Node\_S \*tuya\_ipc\_ring\_buffer\_get\_video\_frame(

CHANNEL\_E channel,

USER\_INDEX\_E userIndex,

BOOL\_T isRetry

);

**Function description**

This interface gets video stream data from the ring buffer.

Note: User indexes distinguish different users in the same ring buffer, and each user maintains the status of getting data.

When getting data lags behind writing data and the lagging value exceeds the threshold, tuya\_ipc\_ring\_buffer\_get\_audio\_frame automatically skips frames to the latest I-Frame or audio frame.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| channel | The ring buffer supports multiple channels that apply to IPC scenarios, such as main streams, substreams, and triple streams. | Input parameter | Yes | See definition of CHANNEL\_E. |
| userIndex | User index | Input parameter | Yes | See definition of USER\_INDEX\_E. |
| isRetry | Indicates whether to retransmit the data | Input parameter | Yes | **0**: No Non-zero value: Yes |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| Ring buffer node data | See section 3.4.1 "Ring\_Buffer\_Node\_S." |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.7.5 tuya\_ipc\_ring\_buffer\_get\_audio\_frame

**Function prototype**

Ring\_Buffer\_Node\_S \*tuya\_ipc\_ring\_buffer\_get\_audio\_frame(

CHANNEL\_E channel,

USER\_INDEX\_E userIndex,

BOOL\_T isRetry);

**Function description**

This interface gets audio stream data from the ring buffer.

Note: User indexes distinguish different users in the same ring buffer, and each user maintains the status of getting data.

When getting data lags behind writing data and the lagging value exceeds the threshold, tuya\_ipc\_ring\_buffer\_get\_audio\_frame automatically skips frames to the latest I-Frame or audio frame.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| channel | The ring buffer supports multiple channels that apply to IPC scenarios, such as audio stream, audio stream 1, and audio stream 2. | Input parameter | Yes | See definition of CHANNEL\_E. |
| userIndex | User index | Input parameter | Yes | See definition of USER\_INDEX\_E. |
| isRetry | Indicates whether to retransmit the data | Input parameter | Yes | **0**: No Non-zero value: Yes |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| Ring buffer node data | See section 3.4.1 "Ring\_Buffer\_Node\_S." |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.7.6 tuya\_ipc\_ring\_buffer\_find\_pre\_video\_by\_frame

**Function prototype**

Ring\_Buffer\_Node\_S \*tuya\_ipc\_ring\_buffer\_find\_pre\_video\_by\_frame(

CHANNEL\_E channel,

UINT\_T frameNum

);

**Function description**

This interface searches for frames based on the **frameNum** value from the latest node to earlier nodes until the I-Frame is found.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| channel | The ring buffer supports multiple channels that apply to IPC scenarios, such as main streams, substreams, and triple streams. | Input parameter | Yes | See definition of CHANNEL\_E. |
| frameNum | Number of frames to find | Input parameter | Yes | Value: 0 to 65535 |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| Ring buffer node data | See section 3.4.1 "Ring\_Buffer\_Node\_S." |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.7.7 tuya\_ipc\_ring\_buffer\_find\_pre\_audio\_by\_frame

**Function prototype**

Ring\_Buffer\_Node\_S \*tuya\_ipc\_ring\_buffer\_find\_pre\_audio\_by\_frame(

CHANNEL\_E channel,

UINT\_T frameNum

);

**Function description**

This interface searches for audio frames based on the **frameNum** value from the latest node to earlier nodes.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| channel | The ring buffer supports multiple channels that apply to IPC scenarios, such as audio stream, audio stream 1, and audio stream 2. | Input parameter | Yes | See definition of CHANNEL\_E. |
| frameNum | Number of frames to find | Input parameter | Yes |  |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| Doorbell buffer node data | See section 3.4.1 "Ring\_Buffer\_Node\_S." |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.7.8 tuya\_ipc\_ring\_buffer\_get\_pre\_video\_frame

**Function prototype**

Ring\_Buffer\_Node\_S \*tuya\_ipc\_ring\_buffer\_get\_pre\_video\_frame(

CHANNEL\_E channel,

USER\_INDEX\_E userIndex,

UINT\_T frameNum

);

**Function description**

This interface searches for frames based on the **frameNum** value from the latest node to earlier nodes until the I-Frame is found. After the pre-recorded data is obtained, tuya\_ipc\_ring\_buffer\_get\_pre\_video\_frame updates the user corresponding to the **userIndex** value on the new node and automatically gets data from the pre-recorded I-Frame.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| channel | The ring buffer supports multiple channels that apply to IPC scenarios, such as main streams, substreams, and triple streams. | Input parameter | Yes | See definition of CHANNEL\_E. |
| userIndex | User index | Input parameter | Yes | See definition of USER\_INDEX\_E. |
| frameNum | Number of frames to find | Input parameter | Yes |  |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| Ring buffer node data | See section 3.4.1 "Ring\_Buffer\_Node\_S." |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.7.9 tuya\_ipc\_ring\_buffer\_get\_pre\_audio\_frame

**Function prototype**

Ring\_Buffer\_Node\_S \*tuya\_ipc\_ring\_buffer\_get\_pre\_audio\_frame(

CHANNEL\_E channel,

USER\_INDEX\_E userIndex,

UINT\_T frameNum

);

**Function description**

This interface searches for audio frames based on the **frameNum** value from the latest node to earlier nodes. After the pre-recorded data is obtained, tuya\_ipc\_ring\_buffer\_get\_pre\_video\_frame updates the user corresponding to the userIndex value on the new node and automatically gets data from the pre-recorded I-Frame.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| channel | The ring buffer supports multiple channels that apply to IPC scenarios, such as audio stream, audio stream 1, and audio stream 2. | Input parameter | Yes | See definition of CHANNEL\_E. |
| userIndex | User index | Input parameter | Yes | See definition of USER\_INDEX\_E. |
| frameNum | Number of frames to find | Input parameter | Yes |  |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| Ring buffer node data | See section 3.4.1 "Ring\_Buffer\_Node\_S." |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.7.10 tuya\_ipc\_ring\_buffer\_anchor\_user\_to\_node

**Function prototype**

VOID tuya\_ipc\_ring\_buffer\_anchor\_user\_to\_node(

CHANNEL\_Echannel,

USER\_INDEX\_E userIndex,

Ring\_Buffer\_Node\_S \*node

);

**Function description**

This interface anchors the user corresponding to a user index on a specified node.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| channel | Audio and video channel | Input parameter | Yes | See definition of CHANNEL\_E. |
| userIndex | User index | Input parameter | Yes | See definition of USER\_INDEX\_E. |
| node | Buffer node data | Input parameter | Yes | See section 3.4.1 "Ring\_Buffer\_Node\_S." |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.7.11 tuya\_ipc\_ring\_buffer\_get\_next\_target\_frame

**Function prototype**

Ring\_Buffer\_Node\_S \*tuya\_ipc\_ring\_buffer\_get\_next\_target\_frame(

CHANNEL\_E channel,

USER\_INDEX\_EuserIndex,

UINT\_T seqNo,

MEDIA\_FRAME\_TYPE\_E type

);

**Function description**

This interface searches for a specified type of node with a larger **seqNo** value from the current position of the user index.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| channel | Audio and video channel | Input parameter | Yes | See definition of CHANNEL\_E. |
| userIndex | User index | Input parameter | Yes | See definition of USER\_INDEX\_E. |
| seqNo |  | Input parameter | Yes |  |
| type | Frame type | Input parameter | Yes | See definition of MEDIA\_FRAME\_TYPE\_E. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| Ring buffer node data | See section 3.4.1 "Ring\_Buffer\_Node\_S." |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.7.12 tuya\_ipc\_ring\_buffer\_clean\_user\_state

**Function prototype**

VOID tuya\_ipc\_ring\_buffer\_clean\_user\_state(

CHANNEL\_E channel,

USER\_INDEX\_E userIndex

);

**Function description**

This interface deletes the status of the ring buffer user corresponding to the **userIndex** value. The tuya\_ipc\_ring\_buffer\_get\_video\_frame or tuya\_ipc\_ring\_buffer\_get\_audio\_frame interface will start from the latest I-Frame.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| channel | Audio and video channel | Input parameter | Yes | See definition of CHANNEL\_E. |
| userIndex | User index | Input parameter | Yes | See definition of USER\_INDEX\_E. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.7.13 FUNC\_REQUEST\_I\_FRAME

**Function prototype**

typedef VOID (\*FUNC\_REQUEST\_I\_FRAME)(

INT\_T channel

);

**Function description**

This is the callback function for applying for I-Frames.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| channel | Video stream channel | Input parameter | Yes | Video stream channels 0, 1, 2, and 3 |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| None | None |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | tuya\_cloud\_types.h  tuya\_cloud\_error\_code.h  tuya\_ipc\_media.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

# 4.8 PTZ Functions

### 4.8.1 tuya\_ipc\_preset\_add

**Function prototype**

OPERATE\_RET tuya\_ipc\_preset\_add(

CHAR\_T \*addr,

UINT\_T size,

S\_PRESET\_POSITION \* preset\_pos

)

**Function description**

This interface adds presets.

Note: The SDK interface does not limit the number of presets, but the Tuya Cloud server supports a maximum of six presets.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| addr | Synchronized image information when a preset is added | Input parameter | No | Set this parameter to null. |
| size | Image size | Input parameter | No | Set this parameter to null. |
| preset\_pos | Position information of an added preset | Input parameter | Yes | See section 3.4.3 "S\_PRESET\_POSITION." The ID in the structure is not configurable and is meaningless. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | preset\_operation.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_types.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.8.2 tuya\_ipc\_preset\_add\_pic

**Function prototype**

OPERATE\_RET tuya\_ipc\_preset\_add\_pic(

CHAR\_T \*addr,

UINT\_T size,

S\_PRESET\_POSITION\* preset\_pos)

**Function description**

This interface adds a preset image.

Note: The tuya\_ipc\_preset\_add\_pic interface adds an image immediately after the tuya\_ipc\_preset\_add interface adds a preset. It cannot be used to add images for multiple presets that have been created in advance.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| addr | Synchronized image information when a preset is added | Input parameter | Yes | Image pointer, supporting the JPG and PNG formats |
| size | Image size | Input parameter | Yes | Image size. The image size is not limited in the SDK and can be adjusted based on network conditions. The recommended size is 150 KB or smaller. |
| preset\_pos | Position information of an added preset | Input parameter | No | The parameter is meaningless from SDK 4.3.2. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | preset\_operation.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_types.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.8.3 tuya\_ipc\_preset\_del

**Function prototype**

OPERATE\_RET tuya\_ipc\_preset\_del(

CHAR\_T\* preset\_id

)

**Function description**

This interface deletes a preset.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| preset\_id | Preset ID registered on the server | Input parameter | Yes | Use the tuya\_ipc\_preset\_get interface to query the preset ID. |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | preset\_operation.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_types.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |

### 4.8.4 tuya\_ipc\_preset\_get

**Function prototype**

OPERATE\_RET tuya\_ipc\_preset\_get (S\_PRESET\_CFG \*preset\_cfg)

**Function description**

This interface queries preset information.

Note: A maximum of six presets can be queried.

**Parameter description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Parameter Type** | **Mandatory** | **Setting** |
| preset\_cfg | Indicates where the returned value is stored after preset configuration information is queried on the server. | Output parameter | Yes | None |

**Returned value**

|  |  |
| --- | --- |
| **Returned Value** | **Description** |
| OPRT\_OK | Success |
| Error code | Error code returned upon a failure |

**Constraints**

|  |  |
| --- | --- |
| **Operating System** | Linux (Ubuntu), LiteOS, and Android |
| **Header File** | preset\_operation.h  tuya\_cloud\_error\_code.h  tuya\_cloud\_types.h |
| **Static Library** | libtuya\_ipc.a  libtuya\_p2p.a |