Interface between Compal cloud and Device/App

Revision:

|  |  |  |
| --- | --- | --- |
| Data | Author | Revision Note |
| 2019/03/20 | Bryan Kuo | V0.1. Initial version |
| 2019/03/21 | Bryan Kuo | V0.2. Add command between device and app |
|  |  |  |
|  |  |  |
|  |  |  |

**Protocol between Compal Cloud and Device/App: HTTP/HTTPS (Cloud is server and Device/App is client)**

**Data Format: JSON**

**Flow:**

1. Device sends “Polling\_Rsp” to Cloud -> Cloud send “CEI\_SN\_Set” with Value to Device -> Device send “ACK\_Rsp” with value to Cloud
2. Device sends “Polling\_Rsp” to Cloud -> Cloud send “CEI\_SN\_Get” to Device -> Device sends “CEI\_SN\_Get” with value to Cloud
3. Cloud sends “CEI\_UID\_Set” with value to Device -> Device sends “ACK\_Rsp” with value to Cloud
4. Cloud sends “CEI\_UID\_Get” to Device -> Device sends “CEI\_UID\_Get” with value to Cloud
5. App sends “Streaming\_Resolution\_Req” with value to Device -> Device sends “Streaming\_Resolution\_Rsp” with value to App

* **Device Settings**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Category** | **Items** | **ID** | **Argument** | **Comment** |
| 1 | Control | Polling\_Rsp |  | NA | Device to Cloud |
| 2 | Control | ACK\_Rsp |  | 1 (means done),  0 (means failed) | Device to Cloud |
| 3 | Device\_Information | CEI\_SN\_Set |  | XXXXXXXXXX | 10 digital, Cloud to Device |
| 4 | Device\_Information | CEI\_SN\_Get |  | NA for request;  XXXXXXXXXX for return | Cloud to Device  10 digital, Device to Cloud |
| 5 | Device\_Information | UID\_Set |  | XXXXXXXXXXXXXXXXXXXX | 20 digital, Cloud to Device |
| 6 | Device\_Information | UID\_Get |  | NA for request;  XXXXXXXXXXXXXXXXXXXX for return | Cloud to Device  20 digital, Device to Cloud |
| 7 | Device\_Information | SW\_Version\_Set |  | DXX.XX.XX | Device to Cloud |
| 8 | Device\_Information | SW\_Version\_Get |  | NA for request;  DXX.XX.XX for return | Cloud to Device  Device to Cloud |
| 9 | Device\_Information | AI\_Version\_Set |  | VXX.XX.XX | Cloud to Device |
| 10 | Device\_Information | AI\_Version\_Get |  | NA for request;  VXX.XX.XX for return | Cloud to Device  Device to Cloud |
| 11 | Device\_Information | Wifi\_MAC\_Set |  | XX:XX:XX:XX:XX:XX | Cloud to Device |
| 12 | Device\_Information | Wifi\_MAC\_Get |  | NA for request;  XX:XX:XX:XX:XX:XX for return | Cloud to Device  Device to Cloud |
| 13 | Device\_Information | Activated\_Get |  | NA for request;  1, 0(Default) | Cloud to Device  Device to Cloud |
| 14 | Device\_Information | DM\_Sync\_Get |  | NA for request;  “CEI\_SN, UID, SW\_Version, AI\_Version, Wifi\_MAC, Activated” for return | Cloud to Device  Device to Cloud |
| 15 | Function | IOTYPE\_USER\_IPCAM\_CEI\_SET\_IR\_ENABLE\_REQ |  | 1(Default), 0 | App to Device |
| 16 | Function | IOTYPE\_USER\_IPCAM\_SPEAKERSTART\_REQ | 0x350 | TBD | TUTK Default Commands  APP to Device |
| 17 | Function | IOTYPE\_USER\_IPCAM\_SPEAKERSTART\_RESP | 0x1111 | typedef struct  {  int result; // 0: success; otherwise: failed.  unsigned char reserved[4];  }SMsgAVIoctrlCEISpeakerStartResp; | Device to APP |
| 18 | Function | IOTYPE\_USER\_IPCAM\_CEI\_SET\_SPK\_ENABLE\_REQ |  | 1, 0(Default) | App to Device |
| 19 | Function | IOTYPE\_USER\_IPCAM\_CEI\_SET\_LSEN\_ENABLE\_REQ |  | 1(Default), 0 | App to Device |
| 20 | Function | IOTYPE\_USER\_IPCAM\_CEI\_SET\_AREADET\_ENABLE\_REQ |  | 1(Default), 0 | App to Device |
| 21 | Function | IOTYPE\_USER\_IPCAM\_CEI\_SET\_SMILEDET\_ENABLE\_REQ |  | 1(Default), 0 | App to Device |
| 22 | Function | IOTYPE\_USER\_IPCAM\_CEI\_SET\_TSEN\_ENABLE\_REQ |  | 1(Default), 0 | App to Device |
| 23 | Function | IOTYPE\_USER\_IPCAM\_CEI\_SET\_HSEN\_ENABLE\_REQ |  | 1(Default), 0 | App to Device |
| 24 | Function | IOTYPE\_USER\_IPCAM\_CEI\_SET\_PICUPLOAD\_ENABLE\_REQ |  | 1(Default), 0 | App to Device |
| 25 | Function | IOTYPE\_USER\_IPCAM\_CEI\_SET\_CRYDET\_ENABLE\_REQ |  | 1(Default), 0 | App to Device |
| 26 | Function | IOTYPE\_USER\_IPCAM\_CEI\_SET\_MIC\_ENABLE\_REQ |  | 1(Default), 0 | App to Device |
| 27 | Function | IOTYPE\_USER\_IPCAM\_CEI\_SET\_LOGUPLOAD\_ENABLE\_REQ |  | 1(Default), 0 | App to Device |
| 28 | Function | IOTYPE\_USER\_IPCAM\_CEI\_SET\_OTA\_ENABLE\_REQ |  | 1(Default), 0 | App to Device |
| 29 | Threshold | IOTYPE\_USER\_IPCAM\_CEI\_SET\_LSEN\_THRES\_REQ |  | XXX | App to Device |
| 30 | Threshold | IOTYPE\_USER\_IPCAM\_CEI\_SETTEMHUM\_REQ | 0x1320 | typedef struct  {  short tem\_enable;  short tem\_low;  short tem\_high;  short hum\_enable;  short hum\_low;  short hum\_high;  unsigned char reserved[4];  } SMsgAVIoctrlCEISetTemperatureHumidity; | App to Device |
| 31 | Threshold | IOTYPE\_USER\_IPCAM\_CEI\_GETTEMHUM\_REQ | 0x1321 | N/A | App to Device |
| 32 | Threshold | IOTYPE\_USER\_IPCAM\_CEI\_GETTEMHUM\_RESP | 0x1322 | same structure as SMsgAVIoctrlCEISetTemperatureHumidity | Device to App |
| 33 | Threshold | IOTYPE\_USER\_IPCAM\_CEI\_SETDANGERZOOM\_REQ (0x1340) |  | typedef struct  {  short enable;  short left;  short top;  short right;  short bottom;  unsigned char reserved[6];  } SMsgAVIoctrlCEISetDangerZoom; | App to Device |
| 34 | Threshold | IOTYPE\_USER\_IPCAM\_CEI\_GETDANGERZOOM\_REQ (0x1341) |  | N/A | App to Device |
| 35 | Threshold | IOTYPE\_USER\_IPCAM\_CEI\_GETDANGERZOOM\_RESP (0x1342) |  | same structure as SMsgAVIoctrlCEISetDangerZoom | Device to App |
| 36 | Control | IOTYPE\_USER\_IPCAM\_CEI\_SETSTREAMCTRL\_REQ | 0x1200 | typedef struct  {  int resolution; //1080(Default), 720, 360  unsigned char reserved[4];  } SMsgAVIoctrlCEISetStreamCtrl; | App to Device |
| 37 | Control | IOTYPE\_USER\_IPCAM\_CEI\_GETSTREAMCTRL\_REQ | 0x1201 | N/A | App to Device |
| 38 | Control | IOTYPE\_USER\_IPCAM\_CEI\_GETSTREAMCTRL\_RESP | 0x1202 | same structure as SMsgAVIoctrlCEISetStreamCtrl | Device to App |
| 39 | Control | IOTYPE\_USER\_IPCAM\_CEI\_MUSIC\_PLAYCONTROL\_REQ | 0x1100 | typedef struct  {  unsigned int channel; // Camera Index  unsigned int command; // play music command. refer to ENUM\_MUSICPLAYCONTROL  unsigned int Param; // command param, that the user defined  STimeDay stTimeDay; // Event time from ListEvent  unsigned char reserved[4];  } SMsgAVIoctrlPlayMusic; | App to Device |
| 40 | Control | IOTYPE\_USER\_IPCAM\_CEI\_MUSIC\_PLAYCONTROL\_RESP (0x1101) | 0x1101 | typedef struct  {  unsigned int command; // Play record command. refer to ENUM\_MUSICPLAYCONTROL  unsigned int result; // Depends on command  // when is AVIOCTRL\_CEI\_MUSIC\_PLAY\_START:  // result>=0 real channel no used by device for playback  // result <0 error  // -1 playback error  // -2 exceed max allow client amount  unsigned char reserved[4];  } SMsgAVIoctrlPlayMusicResp; // only for play record start command | Device to App |
| 41 | Function | IOTYPE\_USER\_IPCAM\_SPEAKERSTOP\_REQ | 0x351 | TBD | TUTK Default Commands  APP to Device |
| 42 | Function | IOTYPE\_USER\_IPCAM\_SPEAKERSTOP\_RESP |  | TBD | Device to App |
| 43 | Function | IOTYPE\_USER\_IPCAM\_START | 0x1FF | TBD | TUTK Default Commands  APP to Device |
| 44 | Function | IOTYPE\_USER\_IPCAM\_STOP | 0x2FF | TBD | TUTK Default Commands  APP to Device |
| 45 | Function | IOTYPE\_USER\_IPCAM\_AUDIOSTART | 0x300 | TBD | TUTK Default Commands  APP to Device |
| 46 | Function | IOTYPE\_USER\_IPCAM\_AUDIOSTOP | 0x301 | TBD | TUTK Default Commands  APP to Device |
| 47 | Function | IOTYPE\_USER\_IPCAM\_RECORD\_PLAYCONTROL | 0x31A | TBD | TUTK Default Commands  APP to Device |
| 48 | Function | IOTYPE\_USER\_IPCAM\_RECORD\_PLAYCONTROL\_RESP | 0x31B | TBD | TUTK Default Commands  APP to Device |

* **App Settings**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Category** | **Items** | **ID** | **Argument** | **Comment** |
| 1 | Control | Polling\_Rsp |  | NA | APP to Cloud |
| 2 | Control | ACK\_Rsp |  | 1 (means done),  0 (means failed) | APP to Cloud |
| 3 | App\_Information | SW\_Version\_Set |  | AXX.XX.XX | APP to Cloud |
| 4 | App\_Information | SW\_Version\_Get |  | NA for request;  AXX.XX.XX for return | Cloud to APP  APP to Cloud |
| 5 | User\_Information | User\_Account\_Set |  | [XXX@XXX.XXX](mailto:XXX@XXX.XXX) | APP to Cloud |
| 6 | User\_Information | User\_Account\_Get |  | NA for request;  [XXX@XXX.XXX](mailto:XXX@XXX.XXX) for return | Cloud to APP  APP to Cloud |
| 7 | User\_Information | Permission\_Set |  | XXX, XXX, XXX, XXX | APP to Cloud |
| 8 | User\_Information | Permission\_Get |  | NA for request;  XXX, XXX, XXX, XXX for return | Cloud to APP  APP to Cloud |

**Device to Compal Cloud Examples**

* DM\_ Sync\_Get

{

"Timestamp":"2019-03-11 21:00:00",

“Device\_Information”:

{

“CEI\_SN\_Get”: “XXXXXXXXXX”,

“UID\_Get”: “XXXXXXXXXXXXXXXXXXXX”,

“SW\_Version\_Get”: “DXX.XX.XX”,

“AI\_Version\_Get”: “VXX.XX.XX”,

“Wifi\_MAC\_Get”: “XX:XX:XX:XX:XX:XX”,

“Activated\_Get:”

}

}

* Polling\_Rsp

{

"Timestamp":"2019-03-11 21:00:00",

“CEI\_SN\_Get”: “XXXXXXXXXX”

“Polling\_Rsp:”NA”

}

* ACK\_Rsp

{

"Timestamp":"2019-03-11 21:00:00",

“CEI\_SN\_Get”: “XXXXXXXXXX”

“ACK\_Rsp:1

}

* CEI\_SN\_Get

{

"Timestamp":"2019-03-11 21:00:00",

“CEI\_SN\_Get”: “XXXXXXXXXX”

}

* UID\_Get

{

"Timestamp":"2019-03-11 21:00:00",

“UID\_Get”: “XXXXXXXXXXXXXXXXXXXX”

}

* Device\_SW\_Version\_Get

{

"Timestamp":"2019-03-11 21:00:00",

“SW\_Version\_Get”: “DXX.XX.XX”

}

* AI\_Version\_Get

{

"Timestamp":"2019-03-11 21:00:00",

“AI\_Version\_Get”: “VXX.XX.XX”

}

* Wifi\_MAC\_Get

{

"Timestamp":"2019-03-11 21:00:00",

“Wifi\_MAC\_Get”: “XX:XX:XX:XX:XX:XX”

}

* Wifi\_MAC\_Get

{

"Timestamp":"2019-03-11 21:00:00",

“Activated\_Get”: “XX:XX:XX:XX:XX:XX”

}

**Compal Cloud to Device Examples**

* DM\_ Sync\_Get

{

"Timestamp":"2019-03-11 21:00:00",

“Device\_Information”:

{

“CEI\_SN\_Get”: “NA”,

“UID\_Get”: “NA”,

“SW\_Version\_Get”: “NA”,

“AI\_Version\_Get”: “NA”,

“Wifi\_MAC\_Get”: “NA”,

“Activated\_Get”:”NA”

}

}

* CEI\_SN\_Set

{

"Timestamp":"2019-03-11 21:00:00",

“CEI\_SN\_Set”: “XXXXXXXXXX”

}

* CEI\_SN\_Get

{

"Timestamp":"2019-03-11 21:00:00",

“CEI\_SN\_Get”: “NA”

}

* UID\_Set

{

"Timestamp":"2019-03-11 21:00:00",

“UID\_Set”: “XXXXXXXXXXXXXXXXXXXX”

}

* UID\_Get

{

"Timestamp":"2019-03-11 21:00:00",

“UID\_Get”: “NA”

}

* Device\_SW\_Version\_Set

{

"Timestamp":"2019-03-11 21:00:00",

“SW\_Version\_Set”: “DXX.XX.XX”

}

* Device\_SW\_Version\_Get

{

"Timestamp":"2019-03-11 21:00:00",

“SW\_Version\_Get”: “NA”

}

* AI\_Version\_Set

{

"Timestamp":"2019-03-11 21:00:00",

“AI\_Version\_Set”: “VXX.XX.XX”

}

* AI\_Version\_Get

{

"Timestamp":"2019-03-11 21:00:00",

“AI\_Version\_Get”: “NA”

}

* Wifi\_MAC\_Set

{

"Timestamp":"2019-03-11 21:00:00",

“Wifi\_MAC\_Set”: “XX:XX:XX:XX:XX:XX”

}

* Wifi\_MAC\_Get

{

"Timestamp":"2019-03-11 21:00:00",

“Activated\_Get”: “NA”

}

**APP to Device Examples**

* Send IOTYPE\_USER\_IPCAM\_CEI\_MUSIC\_PLAYCONTROL\_REQ

SMsgAVIoctrlPlayMusic setPlayMusic = new SMsgAVIoctrlPlayMusic();

byte buf\_info[] = setPlayMusic.parseContent(1, AVIOCTRL\_MUSIC\_PLAY\_START, 100, null);

// Parameter : AVIOCTRL\_MUSIC\_PLAY\_START : start play music

100 : music index 100

int ret = av\_stream\_in.avSendIOCtrl(avIndex, IOTYPE\_USER\_IPCAM\_CEI\_MUSIC\_PLAYCONTROL\_REQ, buf\_info, buf\_info.length);

* Send IOTYPE\_USER\_IPCAM\_CEI\_SETTEMHUM\_REQ

SMsgAVIoctrlCEISetTemperatureHumidity setTemperatureHumidity = new SMsgAVIoctrlCEISetTemperatureHumidity();

byte buf\_info[] = setTemperatureHumidity.parseContent((short) 1, (short) 11, (short) 22, (short) 1, (short) 33, (short) 44);

// Parameter : 1 : enable temperature warning

11 : temperature low

22 : temperature high

1 : enable humidity warning

33 : humidity low

44 : humidity high

int ret = av\_stream\_in.avSendIOCtrl(avIndex, IOTYPE\_USER\_IPCAM\_CEI\_SETTEMHUM\_REQ, buf\_info, buf\_info.length);

* Receive IOTYPE\_USER\_IPCAM\_CEI\_MUSIC\_PLAYCONTROL\_RESP

AVAPIs.avRecvIOCtrl(avIndex, ioType, ioCtrlBuf, MAX\_BUF\_SIZE, 1000);

private void Handle\_IOCTRL\_Cmd(int sid, int avIndex, byte[] buf, int type) {

switch (type) {

case IOTYPE\_USER\_IPCAM\_CEI\_SPEAKERSTART\_RESP: {

SMsgAVIoctrlCEISpeakerStartResp speakerStartResp = new SMsgAVIoctrlCEISpeakerStartResp(buf);

System.out.printf("IOTYPE\_USER\_IPCAM\_CEI\_SPEAKERSTART\_RESP, result = " + speakerStartResp.result + "\n");