Brickcom

BRICKCOM IPCAM HTTP API

1. Preface

This document specifies the Brickcom IPCAM HTTP API which enables applications to access and/or configure the IP Cameras manufactured by Brickcom over a TCP/IP capable network. Developers who wish to write their own utility should follow the API specification herein.

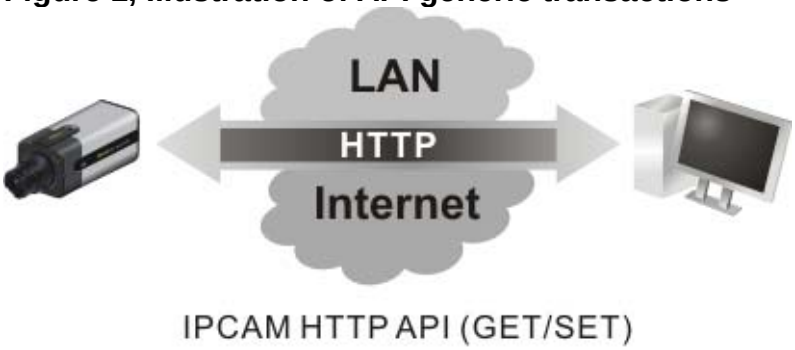
2. TOC

Overview

Brickcom IPCAM HTTP API is the proprietary network control protocol designed by Brickcom Technology to enable applications to access IP Cameras manufactured by Brickcom. The API allows for configuration of the settings and inquiry of current status on these IP Cameras. The API is structured and transmitted over HTTP protocols and hence it is given the name HTTP API.

The complete API is further divided into several categories for ease of management. We dedicate one chapter for each API category to better expound on that API subset.

Figure 1, Illustration of API generic transactions



3. HTTP API Transaction

An HTTP API transaction always starts with a request from a client application. The request is received by the Web server on the IP Camera device and processed by the IP Camera. Finally, it ends with a response which is sent back to the requesting client.

The client HTTP request is taken in two forms:

- HTTP GET: Normally used to retrieve the settings or status of the IP Camera
- HTTP POST: Normally used to configure the settings of the IP Camera

If the request is successfully received by the IP Camera, the response will contain a HTTP header with a 200 OK response code and the HTTP body with the actual response data or other value when error occurs. An example is provided for each request type below:

Illustration 1, Get the network setting from the IP Camera

Client request

```
GET http://<IP Camera address>/network.cgi HTTP/1.0
…
```

Server response

```
HTTP/1.0 200 OK
Content-Type: text/plain

IPAddress=192.168.1.1
SubnetMask=255.255.255.0
...
```

Illustration 2, Set the network setting from the IP Camera

Client request

POST http://<IP Camera address>/**network.cgi** HTTP/1.0

IPAddress=192.168.1.1
SubnetMask=255.255.255.0

Server response

HTTP/1.0 200 OK
…

Error Response

If the IP Camera is unable to handle the client HTTP API request due to certain conditions such as system busy, incorrect parameters, or any other reasons, an appropriate HTTP status code **400 Bad Request** will be returned, and accompanied with an error code and an error string to explain the failure.

Client request

GET/POST …

Server response

HTTP/1.0 200 OK
…
ErrorCode=XXX
ErrorString=Invalid IP Address

4. API Categories

The API categories are listed in the table below.

Table 1, API Categories

API Category	Description
Streaming	Enable users to set/get the setting of multimedia streaming.
Camera	Enable users to set/get the camera/lens setting.
Audio	Enable user to set/get the audio devices setting.
Network	Enable users to set/get the network setting.
Event	Enable users to register to receive notification from IPCAM.
Storage	Enable users to configure storage device for storing media content.
System	Enable users to set/get miscellaneous system settings.
Admin	Enables users to perform administrative tasks over the IP Camera.
Capability	Provide users with the list of available features supported by the IP Camera.
Motion detection	Enable user to set/get the motion detection setting and add/delete/update detection region.
Event	Enable user to set/get the event setting and the notification setting.
I/O control	Enable user to control I/O status

Ps: Fields marked in gray are reserved.

5. Streaming

Streaming API allows applications to
1) set/get the IP Camera streaming setting
2) help users to view video streaming

5.1 getChannels

ActionEvent: getChannels

Request	http://<IP>/cgi-bin/channels.cgi&action=get
Response	size = CH1.index=1 CH1.enabled= CH1.name= CH1.transportType= CH1.video.enabled= CH1.video.format.sourceDevice= CH1.video.format.codecType= CH1.video.format.codecSubType= CH1.video.format.constantBitrate= CH1.video.format.bitrateInKbps= CH1.video.format.resolutionWidth= CH1.video.format.resolutionHeight= CH1.video.format.frameRate= CH1.video.format.gop= CH1.video.format.quality= CH1.video.transport.multicastEnabled= CH1.video.transport.multicastAddress= CH1.video.transport.multicastPort= CH1.video.transport.ttl= CH1.audio.enabled= CH1.audio.format.codecType= CH1.audio.format.codecSubType= CH1.audio.transport.multicastEnabled= CH1.audio.transport.multicastAddress= CH1.audio.transport.multicastPort= CH1.audio.transport.ttl= CH1.meta.enabled= CH1.meta.format.mdAlarmEnabled= CH1.meta.transport.multicastEnabled= CH1.meta.transport.multicastAddress= CH1.meta.transport.multicastPort= CH1.meta.transport.ttl=

	Ch2.index=2 ….
Comment	
Method	GET

5.2 getChannel

ActionEvent: getChannel

Request	http://<IP>/cgi-bin/channels.cgi?action=getChannel&index=<index>
Response	enabled= name= transportType= video.enabled= video.format.codecType= video.format.codecSubType= video.format.constantBitrate= video.format.bitrateInKbps= video.format.resolutionWidth= video.format.resolutionHeight= video.format.frameRate= video.format.gop= video.format.quality= video.transport.multicastEnabled= video.transport.multicastAddress= video.transport.multicastPort= video.transport.ttl= audio.enabled= audio.format.codecType= audio.format.codecSubType= audio.transport.multicastEnabled= audio.transport.multicastAddress= audio.transport.multicastPort= audio.transport.ttl= meta.enabled= meta.format.mdAlarmEnabled= meta.transport.multicastEnabled= meta.transport.multicastAddress= meta.transport.multicastPort= meta.transport.ttl=
Comment	
Method	GET

5.3 addChannel

ActionEvent: addChannel

Request	http://<IP>/cgi-bin/channels.cgi? action=add index=<index> enabled= name= transportType= video.enabled= video.format.codecType= video.format.codecSubType= video.format.constantBitrate= video.format.bitrateInKbps= video.format.resolutionWidth= video.format.resolutionHeight= video.format.frameRate= video.format.gop= video.format.quality= video.transport.multicastEnabled= video.transport.multicastAddress= video.transport.multicastPort= video.transport.ttl= audio.enabled= audio.format.codecType= audio.format.codecSubType= audio.transport.multicastEnabled= audio.transport.multicastAddress= audio.transport.multicastPort= audio.transport.ttl= meta.enabled= meta.format.mdAlarmEnabled= meta.transport.multicastEnabled= meta.transport.multicastAddress= meta.transport.multicastPort= meta.transport.ttl=
Response	
Comment	
Method	POST

5.4 updateChannel

ActionEvent: updateChannel

Request	http://<IP>/cgi-bin/channels.cgi? action=update

	index=<index> enabled= name= transportType= video.enabled= video.format.codecType= video.format.codecSubType= video.format.constantBitrate= video.format.bitrateInKbps= video.format.resolutionWidth= video.format.resolutionHeight= video.format.frameRate= video.format.gop= video.format.quality= video.transport.multicastEnabled= video.transport.multicastAddress= video.transport.multicastPort= video.transport.ttl= audio.enabled= audio.format.codecType= audio.format.codecSubType= audio.transport.multicastEnabled= audio.transport.multicastAddress= audio.transport.multicastPort= audio.transport.ttl= meta.enabled= meta.format.mdAlarmEnabled= meta.transport.multicastEnabled= meta.transport.multicastAddress= meta.transport.multicastPort= meta.transport.ttl=
Response	
Comment	
Method	POST

5.5 updateChannels

ActionEvent: updateChannels

Request	http://<IP>/cgi-bin/channels.cgi? action=updateAll c1Index=& c1Enable=& c1Name=& c1TransportType=& c1VideoEnabled=& c1VideoFormatCodecType=& c1VideoFormatCodecSubType=& c1VideoFormatConstantBitrate=& c1VideoFormatBitrateInKbps =& c1VideoFormatResolutionWidth=& c1VideoFormatResolutionHeight=& c1VideoFormatFrameRate=& c1VideoFormatGop=& c1VideoFormatQuality =& c1VideoTransportMulticastEnabled=& c1VideoTransportMulticastAddress=& c1VideoTransportMulticastPort=& c1VideoTransportTtl=& c1AudioEnabled=& c1AudioFormatCodecType=& c1AudioFormatCodecSubType =& c1AudioTransportMulticastEnabled=& c1AudioTransportMulticastAddress=& c1AudioTransportMulticastPort=& c1AudioTransportTtl=& c1MetaEnabled=& c1MetaFormatMdAlarmEnabled =& c1MetaTransportMulticastEnabled=& c1MetaTransportMulticastAddress=& c1MetaTransportMulticastPort=& c1MetaTransportTtl=& c2Index=& c2Enable=&…….
Response	
Comment	c1VideoFormatGop //default ==0: 1 I-frame/second, ==N: 1 I-frame in N frames
Method	POST

5.6 deleteChannel

ActionEvent: deleteChannel

Request	http://<IP>/cgi-bin/channels.cgi action=delete&index=<index>
Response	
Comment	
Method	POST

SVideoFormatSetting

SVideoFormatSetting	Req or Opt or N	Data type	Allowed Value	notes
sourceDevice	N	Integer	Reserved	**Reserved**
codecType	R	Char[16]	H264, MPEG4, MJPEG,MIMIC, Analytics	
codecSubType	**O**	Char[16]	Depend on encodeType	
constantBitrate	R	Integer	**Table 3**	
bitrateInKbps	R	Integer	**Table 1**	Kbps
quality	R	Integer	[1, 100]	
resolutionWidth	R	Integer	**Table 2**	pixel
resolutionHeight	R	Integer	**Table 2**	pixel
framRate	R	Integer	[1, 30]	HZ Based on the limit of hardware
gop	R	Integer	Integer	default =>0: 1 I-frame/second, =>N: 1 I-frame in N frames

Note: When **constantBitrate** =0, only **quality** can be set; **constantBitrate** = 1, **bitrateInKbps** can be set.

Table 1: Available bitrates

bitrateInKbps
64
128
256
384
512
768
1,500
2,000
4,000
6,000
8,000
10,000
12,000
15,000
20,000

Table 2: Available resolutions

resolutionWidth	resolutionHeight	Input module
1280	800	OV9710、OV9715
640	400	OV9710、OV9715
320	192	OV9710、OV9715
1280	1024	MT9M131
640	512	MT9M131
320	256	MT9M131
720	480	TW9910
352	240	TW9910

Table 3: Available constantBitrate

constantBitrate	Value
VBR	0
CBR	1

SAudioFormatSetting

SAudioFormatSetting	Req or Opt or N	Data type	Allowed Value	notes
sourceDevice	N	Integer	Reserved	
codecType	R	Char[16]	**table 3**	
codecSubType	R	Char[16]	**table 3**	
numberOfChannel	**N**	Integer	[0,1]	**Reserved** 0: Mono 1:Stereo
sampleRate	**N**	Integer		**HZ Reserved**
frameIntervalMS	**N**	Integer		**MS Reserved**
sampleSizeBit	**N**	Integer		**Reserved**

Table 3: Available codeType and codeSubType

codeType	codeSubType	note
G711	AUTO	default is G.711 mu-law
G711	PCMU	G.711 mu-law
G711	PCMA	G.711 A-law
G726	AUTO	default is G.726 ADPCM at 32 kbps
G726	G726-16	G.726 ADPCM at 16 kbps
G726	G726-24	G.726 ADPCM at 24 kbps
G726	G726-32	G.726 ADPCM at 32 kbps
G726	G726-40	G.726 ADPCM at 40 kbps
AMR	AUTO	default is AMR at 12.2 kbps
AMR	AMR-MR475	AMR at 4.75 kbps
AMR	AMR-MR515	AMR at 5.15 kbps
AMR	AMR-MR59	AMR at 5.9 kbps
AMR	AMR-MR67	AMR at 6.7 kbps
AMR	AMR-MR74	AMR at 7.4 kbps
AMR	AMR-MR795	AMR at 7.95 kbps
AMR	AMR-MR102	AMR at 10.2 kbps
AMR	AMR-MR122	AMR at 12.2 kbps

STransportSetting (Video/Audio)

STransportSetting	Req or Opt or N	Data type	Allowed Value	notes
multicastEnabled	R	Int	[0,1]	0:disabled 1:enabled
multicastAddress	R	Char[16]	232.0.1.0-232.255.255.255	RFC4607
multicastPort	R	Int	[1025,65534]	
ttl	R	Int	[1,255]	

SMetaFormatSetting

SMetaFormatSetting	Req or Opt or N	Data type	Allowed Value	notes
mdAlarmEnabled	R	Int	[0,1]	0:disabled 1:enabled

SVideoSessionSetting

SVideoSessionSetting	Req or Opt or N	Data type	Allowed Value	notes
enabled	R	Int	[0,1]	0:disabled 1:enabled

SAudioSessionSetting

SAudioSessionSetting	Req or Opt or N	Data type	Allowed Value	notes
enabled	R	Int	[0,1]	0:disabled 1:enabled

SMetaSessionSetting

SMetaSessionSetting	Req or Opt or N	Data type	Allowed Value	notes
enabled	R	Int	[0,1]	0:disabled 1:enabled

SChannelSetting

SChannelSetting	Req or Opt or N	Data type	Allowed Value	notes
enabled	R	Int	[0,1]	0:disabled 1:enabled
Index	R	Int	[0,5]	0**:Reserved**, Sequence number
Name	**O**	Char[16]	**Table 4**	Reject []<>"#'\&|;?*SPACE
transportType	R	Int	**Table 5**	

Table 4: Available **name**

name
H264
MJPEG
MPEG4
MSN
Analytics

Table 5: Available **transportType**

transportType	Value
TRANSPORT_TYPE_RTSP_RTP	**0**
TRANSPORT_TYPE_RTP_ONLY	**1**
TRANSPORT_TYPE_HTTP	**2**
TRANSPORT_TYPE_MSN	**3**
TRANSPORT_TYPE_ANALYTICS	**4**

5.7 getStream

ActionEvent: getStream

Request	rtsp://<IP>/channel<index>
Response	
Comment	<Index> is the index number of the SChannelSetting. If user enables MJPG over http setting, user can set request http://<IP>/channel<index> For example:rtsp://192.168.1.100/channel1
Method	

5.8 setSnapshot

ActionEvent: setSnapshot

Request	http://<IP>/cgi-bin/media.cgi? action=**setSnapshot** resolution=1280x800
Response	
Comment	**Cannot dynamically set the resolution.**
Method	POST

5.9 getSnapshot

ActionEvent: getSnapshot

Request	http://<IP>/cgi-bin/media.cgi?action=getSnapshot
Response	
Comment	Use this API to get the picture snapshot of current vision.
Method	GET

5.10 setRtsp

ActionEvent: setRtsp

Request	http://<IP>/cgi-bin/ rtsp.cgi? action=set port=554 authentication=[0,1,2]
Response	
Comment	
Method	POST

SRTSPSetting	Req or Opt or N	Data type	Allowed Value	notes
port	R	Int	554, [1025,65534]	
authentication	R	Int	[0,2]	0: NONE 1: BASIC 2: DIGEST

5.11 getRtsp

ActionEvent: getRtsp

Request	http://<IP>/cgi-bin/rtsp.cgi?action=get
Response	rtsp.port=554 rtsp.authentication=
Comment	
Method	GET

5.12 getinboundChannel

ActionEvent: getinboundChannel

Request	http://<IP>/cgi-bin/ inboundChannel.cgi?action=get
Response	ch1.transportType=1 ch1.port=12345 ch1.codec=PCMA ch2.transportType= ch2.port ch2.codec
Comment	
Method	GET

6. Camera

Camera API allows applications to set/get the Camera/lens setting.

6.1 setWhiteBalance

ActionEvent: setWhiteBalance

Request	http://<IP>/cgi-bin/camera.cgi? action=**setWhiteBalance** mode=
Response	
Comment	
Method	POST

SWhiteBalanceSetting	Req or Opt or N	Data type	Allowed Value	notes
mode	N	Int	[0,2]	WB_MODE_OFF=0 WB_MODE_SIMPLE=1 WB_MODE_ADVANCED=2
level	N	Int	[0,100]	**Reserved**

6.2 getWhiteBalance

ActionEvent: getWhiteBalance

Request	http://<IP>/cgi-bin/camera.cgi?action=**getWhiteBalance**
Response	mode=
Comment	
Method	GET

6.3 setBrightness

ActionEvent: setBrightness

Request	http://<IP>/cgi-bin/camera.cgi? action= **setBrightness**

	level=
Response	
Comment	
Method	POST

SBrightnessSetting	Req or Opt or N	Data type	Allowed Value	notes
level	R	Int	[-5,5]	Based on the limit of hardware

6.4 getBrightness

ActionEvent: getBrightness

Request	http://<IP>/cgi-bin/camera.cgi?action=**getBrightness**
Response	level=
Comment	
Method	GET

6.5 setColorSaturation

ActionEvent: setColorSaturation

Request	http://<IP>/cgi-bin/camera.cgi? action= **setColorSaturation** level=
Response	
Comment	
Method	POST

SColorSaturationSetting	Req or Opt or N	Data type	Allowed Value	notes
level	R	Int	[-5,5]	Based on the limit of hardware

6.6 getColorSaturation

ActionEvent: getColorSaturation

Request	http://<IP>/cgi-bin/camera.cgi?action=**getColorSaturation**
Response	level=
Comment	
Method	GET

6.7 setMirrorFlip

ActionEvent: setMirrorFlip

Request	http://<IP>/cgi-bin/camera.cgi? action= **setMirrorFlip** mirrorEnabled = flipEnabled=
Response	
Comment	
Method	POST

SMirrorFlipSetting	Req or Opt or N	Data type	Allowed Value	notes
mirrorEnabled	R	Int	[0,1]	0:disabled; 1:enabled
flipEnabled	R	Int	[0,1]	0:disabled; 1:enabled

6.8 getMirrorFlip

ActionEvent: getMirrorFlip

Request	http://<IP>/cgi-bin/camera.cgi?action= **getMirrorFlip**
Response	flipEnabled= mirrorEnabled =
Comment	
Method	GET

6.9 setSharpness

ActionEvent: setSharpness

Request	http://<IP>/cgi-bin/camera.cgi? action= **setSharpness** level=
Response	
Comment	
Method	POST

SSharpnessSetting	Req or Opt or N	Data type	Allowed Value	notes
level	R	Int	[-5,5]	Based on the limit of hardware

6.10 getSharpness

ActionEvent: getSharpness

Request	http://<IP>/cgi-bin/camera.cgi?action=**getSharpness**
Response	level=
Comment	
Method	GET

6.11 setContrast

ActionEvent: setContrast

Request	http://<IP>/cgi-bin/camera.cgi? action=**setContrast** level=
Response	
Comment	
Method	POST

SContrastSetting	Req or Opt or N	Data type	Allowed Value	notes
level	R	Int	[-5,5]	Based on the limit of hardware

6.12 getContrast

ActionEvent: getContrast

Request	http://<IP>/cgi-bin/camera.cgi?action=**getContrast**
Response	level=
Comment	
Method	GET

6.13 setFrequcny

ActionEvent: setFrequcny

Request	http://<IP>/cgi-bin/camera.cgi? action=**setFrequency** freq =
Response	
Comment	
Method	POST

SFrequencySetting	Req or Opt or N	Data type	Allowed Value	notes
freq	R	Int	[0,1]	0, FREQ_60HZ 1, FREQ_50HZ

6.14 getFrequency

ActionEvent: getFrequency

Request	http://<IP>/cgi-bin/camera.cgi?action=**getFrequency**
Response	freq=
Comment	
Method	GET

6.15 setEffect

ActionEvent: setEffect

Request	http://<IP>/cgi-bin/camera.cgi? action=**setEffect** effectMode = colorMode=
Response	
Comment	
Method	POST

SEffectSetting	Req or Opt or N	Data type	Allowed Value	notes
effectMode	R	Int	[0,1]	0: Auto 1: Manual
colorMode	R	Int	[0,1]	0: Color 1: Black&White

6.16 getEffect

ActionEvent: getEffect

Request	http://<IP>/cgi-bin/camera.cgi?action=**getEffect**
Response	effectMode= colorMode=
Comment	
Method	GET

6.17 setEnvMode

ActionEvent: setEnvMode

Request	http://<IP>/cgi-bin/camera.cgi? action=**setEnvMode** envMode =
Response	
Comment	
Method	POST

SEnvModeSetting	Req or Opt or N	Data type	Allowed Value	notes
envMode	R	Int	[0,1]	0,MODE_OUTDOOR 1,MODE_INDOOR

6.18 getEnvMode

ActionEvent: getEnvMode

Request	http://<IP>/cgi-bin/camera.cgi?action=**getEnvMode**
Response	envMode=
Comment	
Method	GET

6.19 setIRCutFilter

ActionEvent: setIRCutFilter

Request	http://<IP>/cgi-bin/camera.cgi? action=**setIRCutFilter** mode= thresholdLevel=
Response	
Comment	
Method	POST

SIRCutFilterSetting	Req or Opt or N	Data type	Allowed Value	notes
mode	R	Int	[0,5]	0, IRCUT_MODE_OFF 1,IRCUT_MODE_ON 2,IRCUT_MODE_AUTO
thresholdLevel	R	Int	[0,100]	**Reserved**

6.20 getIRCutFilter

ActionEvent: getIRCutFilter

Request	http://<IP>/cgi-bin/camera.cgi?action=**getIRCutFilter**
Response	mode= thresholdLevel=
Comment	
Method	GET

6.21 setIRLED

ActionEvent: setIRLED

Request	http://<IP>/cgi-bin/camera.cgi? action=**setIRLED** mode= thresholdLevel=
Response	
Comment	
Method	POST

SIRLEDSetting	Req or Opt or N	Data type	Allowed Value	notes
mode	R	Int	[0,2]	0, IRLED_OFF 1,IRLED_ON 2,IRLED_MODE_AUTO
thresholdLevel	R	Int	[0,100]	**Reserved**

6.22 getIRLED

ActionEvent: getIRLED

Request	http://<IP>/cgi-bin/camera.cgi?action=**getIRLED**
Response	mode= thresholdLevel=
Comment	
Method	GET

6.23 setVideoOverlay

ActionEvent: setVideoOverlay

Request	http://<IP>/cgi-bin/camera.cgi? action=**setVideoOverlay** useTimestamp= displayString= useImage= useText= osdPalette1.y= osdPalette1.Cb= osdPalette1.Cr= osdPalette2.y= osdPalette2.Cb= osdPalette2.Cr= osdWindow1.x= osdWindow1.y= osdWindow1.transparent= osdWindow2.x= osdWindow2.y= osdWindow2.transparent=
Response	
Comment	
Method	POST

SOsdPalette	Req or Opt or N	Data type	Allowed Value	notes
y	R	Int	[0,255]	
Cb	R	Int	[0,255]	
Cr	R	Int	[0,255]	

SOsdWindow	Req or Opt or N	Data type	Allowed Value	notes
x	R	Int		Range:depends on resolution
y	R	Int		Range:depends on resolution
transparent	R	Int	[0,3]	0:0%, 1:50%, 2:75%, 3:100%

SVideoOverlaySetting	Req or Opt or N	Data type	Allowed Value	notes
useTimestamp	R	Int	[0,1]	0: TimeStamp_off 1: TimeStamp_on
displayString	R	Char[50]	[0,1]	
useImage	R	Int	[0,1]	
enabled	R	Int	[0,1]	0:disabled 1:enabled

6.24 getVideoOverlay

ActionEvent: getVideoOverlay

Request	http://<IP>/cgi-bin/camera.cgi?action=**getVideoOverlay**
Response	useTimestamp= displayString= useImage= useText= osdPalette1.y= osdPalette1.Cb= osdPalette1.Cr=

	osdPalette2.y= osdPalette2.Cb= osdPalette2.Cr= osdWindow1.x= osdWindow1.y= osdWindow1.transparent= osdWindow2.x= osdWindow2.y= osdWindow2.transparent=
Comment	
Method	GET

6.25 setAutoIris

ActionEvent: setAutoIris

Request	http://<IP>/cgi-bin/camera.cgi? action= **setAutoIris** enabled
Response	
Comment	
Method	POST

SAutoIris	Req or Opt or N	Data type	Allowed Value	notes
enabled	R	Int	[0,1]	0:DISABLED 1:ENABLED

6.26 getAutoIris

ActionEvent: getAutoIris

Request	http://<IP>/cgi-bin/camera.cgi?action= **getAutoIris**
Response	enabled=
Comment	
Method	GET

6.27 setCameraSetting

ActionEvent: setCameraSetting

Request	http://<IP>/cgi-bin/camera.cgi? action=**setCameraSetting** whiteBalance.mode=0 whiteBalance.level=0 brightness.level=1 colorSaturation.level=-1 flipEnabled=0 mirrorEnabled=0 sharpness.level=2 contrast.level=0 freq=0 effectMode=0 colorMode= envMode=1 IRCutFilter.mode=2 IRCutFilter.thresholdLevel=0 IRLED.mode=2 IRLED.thresholdLevel=0 autoIris.enabled=1 videoOverlay.useTimestamp=1 videoOverlay.displayString=HELLO videoOverlay.useImage=0 videoOverlay.useText= videoOverlay.osdPalette1.y=255 videoOverlay.osdPalette1.Cb=128 videoOverlay.osdPalette1.Cr=128 videoOverlay.osdPalette2.y=16 videoOverlay.osdPalette2.Cb=128 videoOverlay.osdPalette2.Cr=128 videoOverlay.osdWindow1.x=0 videoOverlay.osdWindow1.y=13 videoOverlay.osdWindow1.transparent=0 videoOverlay.osdWindow2.x=0 videoOverlay.osdWindow2.y=0 videoOverlay.osdWindow2.transparent=0 shutterSpeed.level=0 gain.mode= gain.level=0 privacy.enabled=0 privacy.FullScreenEnabled=0 privacy.PriWindow1.x=0 privacy.PriWindow1.y=0 privacy.PriWindow1. enabled=0 privacy.PriWindow2.x=0 privacy.PriWindow2.y=0

	privacy.PriWindow2. enabled=0
Response	
Comment	
Method	POST

6.28 getCameraSetting

ActionEvent: getCameraSetting

Request	http://<IP>/cgi-bin/camera.cgi?action=**getCameraSetting**
Response	whiteBalance.mode=0 brightness.level=1 colorSaturation.level=-1 flipEnabled=0 mirrorEnabled=0 sharpness.level=2 contrast.level=0 freq=0 effectMode=0 colorMode= envMode=1 IRCutFilter.mode=2 IRCutFilter.thresholdLevel=0 IRLED.mode=2 IRLED.thresholdLevel=0 autoIris.enabled=1 videoOverlay.useTimestamp=1 videoOverlay.displayString=HELLO videoOverlay.useImage=0 videoOverlay.useText= videoOverlay.osdPalette1.y=255 videoOverlay.osdPalette1.Cb=128 videoOverlay.osdPalette1.Cr=128 videoOverlay.osdPalette2.y=16 videoOverlay.osdPalette2.Cb=128 videoOverlay.osdPalette2.Cr=128 videoOverlay.osdWindow1.x=0 videoOverlay.osdWindow1.y=13 videoOverlay.osdWindow1.transparent=0 videoOverlay.osdWindow2.x=0 videoOverlay.osdWindow2.y=0 videoOverlay.osdWindow2.transparent=0 shutterSpeed.level=0 gain.mode=0 gain.level=0 privacy.enabled=0 privacy.FullScreenEnabled=0 privacy.PriWindow1.x=0 privacy.PriWindow1.y=0 privacy.PriWindow1. enabled=0

	privacy.PriWindow2.x=0 privacy.PriWindow2.y=0 privacy.PriWindow2. enabled=0
Comment	
Method	GET

6.29 setShutterSpeed

ActionEvent: setShutterSpeed

Request	http://<IP>/cgi-bin/camera.cgi? action= **setShutterSpeed** level=
Response	
Comment	
Method	POST

ShutterSpeed	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
level	R	Int	[0,4]	

6.30 getShutterSpeed

ActionEvent: getShutterSpeed

Request	http://<IP>/cgi-bin/camera.cgi?action=**getShutterSpeed**
Response	level=
Comment	
Method	GET

6.31 setGain

ActionEvent: setGain

Request	http://<IP>/cgi-bin/camera.cgi action= **setGain** mode= level=
Response	
Comment	
Method	POST

Gain	Req or Opt or N	Data type	Allowed Value	notes
mode	R	Int	[0,1]	0:disabled 1:enabled
level	R	Int	[0,4]	

6.32 getGain

ActionEvent: getGain

Request	http://<IP>/cgi-bin/camera.cgi?action=**getGain**
Response	mode= level=
Comment	
Method	GET

6.33 setPrivacy

ActionEvent: setPrivacy

Request	http://<IP>/cgi-bin/camera.cgi? action= **setPrivacy** enabled= fullScreenEnabled= privacyButtonEnabled= priWindow1. enabled= priWindow1.x1= priWindow1.y1= priWindow1.x2= priWindow1.y2= priWindow2. enabled= priWindow2.x1= priWindow2.y1= priWindow2.x2= priWindow2.y2=
Response	
Comment	
Method	POST

Privacy	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
enabled	R	Int	[0,1]	0:disabled 1:enabled Feature works.
fullScreenEnabled	R	Int	[0,1]	0:disabled 1:enabled
privacyButtonEnabled	R	Int	[0,1]	0:disabled 1:enabled Function works.

PriWindow	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
enabled	R	Int	[0,1]	0:disabled 1:enabled
x1	R	Int		depends on resolution
x2	R	Int		depends on resolution

y1	R	Int		depends on resolution
y2	R	Int		depends on resolution

6.34 getPrivacy

ActionEvent: getPrivacy

Request	http://<IP>/cgi-bin/camera.cgi action= **getPrivacy**
Response	enabled= fullScreenEnabled= privacyButtonEnabled= priWindow1. enabled= priWindow1.x1= priWindow1.y1= priWindow1.x2= priWindow1.y2= priWindow2. enabled= priWindow2.x1= priWindow2.y1= priWindow2.x2= priWindow2.y2=
Comment	
Method	POST

7. Audio

Audio API allows applications to
1) set/get the audio device setting
2) set/get the volume of the audio device

7.1 setAudioDevice

ActionEvent: setAudioDevice

Request	http://<IP>/cgi-bin/audio.cgi? action=**setAudioDevice** muted= level = mediaType= voiceSource = echo=
Response	
Comment	
Method	POST

SAudioDeviceSetting	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
muted	R	Int	[0,1]	1: (muted), 0: (un-muted)
level	R	Int	[1,100]	Speaker volume
mediaType	N	Int	[0,1]	0,Full 1, Half duplex
voiceSource	R	Int	[0,1]	0, MIC 1, Line in
echo	R	Int	[0,1]	0:disabled 1:enabled

7.2 getAudioDevice

ActionEvent: getAudioDevice

Request	http://<IP>/cgi-bin/ audio.cgi?action=**getAudioDevice**
Response	muted = level = mediaType= voiceSource =
Comment	
Method	GET

7.3 setAudioMuteState

ActionEvent: setAudioMuteState

Request	http://<IP>/cgi-bin/audio.cgi? action=**setAudioMuteState** muted=
Response	
Comment	
Method	POST

7.4 getAudioMuteState

ActionEvent: getAudioMuteState

Request	http://<IP>/cgi-bin/audio.cgi?action=**getAudioMuteState**
Response	muted=
Comment	
Method	GET

7.5 setAudioVolume

ActionEvent: setAudioVolume

Request	http://<IP>/cgi-bin/audio.cgi? action=**setAudioVolume** level=
Response	
Comment	
Method	POST

7.6 getAudioVolume

ActionEvent: getAudioVolume

Request	http://<IP>/cgi-bin/audio.cgi?action=**getAudioVolume**
Response	level=
Comment	
Method	GET

7.7 playaudio

ActionEvent: play

Request	http://<IP>/cgi-bin/audio.cgi? action=play name=
Response	
Comment	The device will start to play the audio
Method	POST

	Req or Opt or N	Data type	Allowed Value	notes
name	R	Char[64]		

7.8 stopaudio

ActionEvent: stopaudio

Request	http://<IP>/cgi-bin/audio.cgi? action=stopaudio
Response	
Comment	The device will stop playing the audio
Method	POST

7.9 recordaudio

ActionEvent: record

Request	http://<IP>/cgi-bin/audio.cgi? action=record name=
Response	
Comment	The device will start to record the audio
Method	POST

7.10 stoprecordaudio

ActionEvent: stoprecord

Request	http://<IP>/cgi-bin/audio.cgi? action=stoprecord
Response	
Comment	The device will stop recording the audio
Method	POST

7.11 getFilestatus

ActionEvent: getFilestatus

Request	http://<IP>/cgi-bin/audio.cgi?action=**getFilestatus**
Response	size= audiofile1.name= audiofile1.size= audiofile1.time= audiofile1.codecType= audiofile2.name= audiofile2.size= audiofile2.time= audiofile2.codecType= ……
Comment	
Method	GET

7.12 removeAudioFile

ActionEvent: remove

Request	http://<IP>/cgi-bin/audio.cgi? action=remove name=
Response	
Comment	The device will stop recording the audio
Method	POST

8. Network

Network API allows applications to set/get the network-related settings including IP address, WIFI network, etc.

8.1 setBasicNetwork

ActionEvent: setBasicNetwork

Request	http://<IP>/cgi-bin/basicNetwork.cgi?
	action= **set**
	//STATIC
	addressType=0
	ipv4Address=
	subnetMask=
	gatewayAddress=
	dnsAddress1=
	dnsAddress2=
	// DHCP,
	addressType=1
	// PPPOE
	addresssType=2
	pppoe.username=
	pppoe.password=

Response	
Comment	
Method	POST

SPPPoESetting	Req or Opt or N	Data type	Allowed Value	notes
username	R	Char[128]		Reject ^[`#$^&*{}[]\";\'\|\/<>?\\ ]$%
password	R	Char[128]		reject []\&|:";<>?,/+=*'%#space

SBasicNetworkSetting	Req or Opt or N	Data type	Allowed Value	notes
addressType	R	Int	[0,2]	0, STATIC 1, DHCP 2, PPPOE

ipv4Address	R	Char[16]		1~223 . 0~255 . 0~255 . 1~254 reject: 127.0.0.1
subnetMask	**R**	Char[16]		1~255 . 0~255 . 0~255 . 0~254
gatewayAddress	**R**	Char[16]		1~223 . 0~255 . 0~255 . 1~254 reject: 127.0.0.1
dnsAddress1	**R**	Char[16]		1~223 . 0~255 . 0~255 . 1~254 reject: 127.0.0.1
dnsAddress2	**R**	Char[16]		1~223 . 0~255 . 0~255 . 1~254 reject: 127.0.0.1

8.2 getBasicNetwork

ActionEvent: getBasicNetwork

Request	http://<IP>/cgi-bin/basicNetwork.cgi?action=**get**
Response	addressType= (0=Static,1=DHCP, 2=PPPoE) ipv4Address= subnetMask= gatewayAddress= dnsAddress1= dnsAddress2= pppoe.username= pppoe.password=
Comment	
Method	GET

8.3 setUPnP

ActionEvent: setUPnP

Request	http://<IP>/cgi-bin/upnp.cgi? action=**set** enabled= name=
Response	
Comment	
Method	POST

SUPnPSetting	Req or Opt or N	Data type	Allowed Value	notes
enabled	R	Int	[0,1]	0:disabled 1:enabled

upnpName	R	Char[128]		

8.4 getUPnP

ActionEvent: getUPnP

Request	http://<IP>/cgi-bin/upnp.cgi?action=**get**
Response	enabled= name=
Comment	
Method	GET

8.5 setDDNS

ActionEvent: setDDNS

Request	http://<IP>/cgi-bin/ddns.cgi? action=set dyndnsEnabled= dyndns.wildcardEnabled= dyndns.username= dyndns.password= dyndns.hostname= tzodnsEnabled= tzodns.wildcardEnabled= tzodns.username= tzodns.password= tzodns.hostname=
Response	
Comment	
Method	POST

SDDNSSetting	Req or Opt or N	Data type	Allowed Value	notes
dyndnsEnabled	R	Int	[0,1]	0:disabled 1:enabled
tzodnsEnabled	R	Int	[0,1]	0:disabled 1:enabled

SDDNSEntry	Req or Opt or N	Data type	Allowed Value	notes
username	R	Char[128]		Table
password	R	Char[128]		Table
hostname	R	Char[128]		
wildcardEnabled	R	Int	[0,1]	0:disabled 1:enabled

Table: username

SDDNSEntry	RejectRule
dyndns	Reject[]\&|"; >*`'#$^.?/{}
tzodns	Should be EmailAddress formet type.

Table:password

SDDNSEntry	Length	RejectRule
dyndns	6~29	Reject[]\&|:";<>?,/+=*'%#space
tzodns	16~24	Reject[]\&|:";<>?,/+=*'%#space

8.6 getDDNS

ActionEvent: getDDNS

Request	http://<IP>/cgi-bin/ddns.cgi? action=**get**
Response	dyndnsEnabled=0 dyndns.wildcardEnabled= dyndns.username= dyndns.password= dyndns.hostname= tzodnsEnabled= tzodns.wildcardEnabled= tzodns.username= tzodns.password= tzodns.hostname=
Comment	
Method	GET

8.7 setEthernet

ActionEvent: setEthernet

Request	http://<IP>/cgi-bin/ethernet.cgi? action=**set** mediaType=
Response	
Comment	
Method	POST

SEthernetSetting	Req or Opt or N	Data type	Allowed Value	notes
mediaType	R	Int	[0,4]	0, AUTO 1, 10_HALF_DUPLEX 2, 10_FULL_DUPLEX 3, 100_HALF_DUPLEX 4, 100_FULL_DUPLEX

8.8 getEthernet

ActionEvent: getEthernet

Request	http://<IP>/cgi-bin/ethernet.cgi?action=**get**
Response	mediaType=
Comment	
Method	GET

8.9 setWIFI

ActionEvent: setWIFI

Request	http://<IP>/cgi-bin/wifi.cgi? action=**set** enabled= mode= operationMode= channel= SSID= preamble= rtsThreshold= fragmentationThreshold= authentication= channelBandWidth= securityMode= WEP. authenticationType= WEP. defaultTransmitKeyIndex = WEP. wepKeyLength = WEP. encryptionKeyList. Keyentry1.encryptionKey= WEP. encryptionKeyList. Keyentry2.encryptionKey= WEP. encryptionKeyList. Keyentry3.encryptionKey= WEP. encryptionKeyList. Keyentry4.encryptionKey= WPA. algorithmType= WPA.sharedKey= WPS.WPSMode= WPS.PINCode=
Response	
Comment	
Method	POST

SWPS	Req or Opt or N	Data type	Allowed Value	notes
WPSMode	R	Int	[0,2]	0:NONE 1:PIN 2:PBC
PINCode	R	Char[64]		

SWPA	Req or Opt or N	Data type	Allowed Value	notes
algorithmType	R	Int	[0,3]	0: TKIP 1: AES 2: TKIP_AES
sharedKey	R	Char[64]		

SKeyentry	Req or Opt or N	Data type	Allowed Value	notes
encryptionKey	R	Char[64]		

SWEP	Req or Opt or N	Data type	Allowed Value	notes
authenticationType	R	Int	[0,2]	0: OPEN 1: SHARED 2: WEPAUTO
defaultTransmitKeyIndex	R	Int	[1,4]	
wepKeyLength	R	Int		Key length:64 bits or 128 bits

SWIFISetting	Req or Opt or N	Data type	Allowed Value	notes
enabled	R	Int	[0,1]	0:disabled 1:enabled
mode	R	Int	[0,1]	0,WIFI_ACCESS_MODE_INFRASTRUCTURE 1,WIFI_ACCESS_MODE_ADHOC
operationMode	**R**	Int	[0,6]	0,WIFI_OP_MODE_AU

				TO 1,WIFI_OP_MODE_11G_ONLY 2,WIFI_OP_MODE_11B_ONLY 3, WIFI_OP_MODE_11N_ONLY 4, WIFI_OP_MODE_11BG_MIXED 5, WIFI_OP_MODE_11GN_MIXED 6, WIFI_OP_MODE_11BGN_MIXED
channel	**R**	Int	**[0,13]**	(0) Auto, 1-13
wmm	**R**	Int		0:disabled 1:enabled
SSID	**R**	Char[31]		
Preamble	**R**	Int	**[0,1]**	0,WIFI_PREAMBLE_TYPE_LONG 1,WIFI_PREAMBLE_TYPE_SHORT
rtsThreshold	**R**	Int	**[1,2347]**	
fragmentationThreshold	**R**	Int	**[256,2346]**	
authentication	**R**	Int	**[0,1]**	0,WIFI_AUTHENTICATION_TYPE_OPEN 1,WIFI_AUTHENTICATION_TYPE_SHARED_KEY
channelBandWidth	**R**	Int	**[0,1]**	0, FORTY_MHZ 1, TWENTY_MHZ
securityMode	**R**	I nt	**[0,3]**	0, WL_NONE 1, WL_WEP 2, WL_WPAPSK 3, WL_WPA2PSK

SIEEE_802_1xSetting	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
enabled	R	Int	[0,1]	0:disabled 1:enabled
authenticationProtocolType	R	Int	[0,4]	Table

innerTTLSAuthenticationMethod	R	Int	[0,3]	Table
innerEAPProtocolType	R	Int	[0,1]	Table
validateServerEnabled	R	Int	[0,1]	0:disabled 1:enabled
userName	R	Char[65]		
password	R	Char[65]		
anonymousID	R	Char[65]		
autoPACProvisioningEnabled	R	Int	[0,1]	0:disabled 1:enabled
caline	R	Int		
clientline	R	Int		
PACline	R	Int		

Table: Available **authenticationProtocolType**

type	Value
WL_EAP_TLS	**0**
WL_EAP_TTLS	**1**
WL_EAP_PEAP	**2**
WL_EAP_FAST	**3**
WL_EAP_LEAP	**4**

Table: Available **innerTTLSAuthenticationMethod**

type	Value
WL_MSCHAP	**0**
WL_MSCHAPV2	**1**
WL_PAP	**2**
WL_EAP_FAST	**3**

Table: Available **innerEAPProtocolType**

type	Value
WL_INNER_EAP_TLS	**0**
WL_EAP_OTP	**1**

8.10 getWIFI

ActionEvent: getWIFI

Request	http://<IP>/cgi-bin/wifi.cgi? action=**get**
Response	enabled= mode= operationMode= channel= SSID= preamble= rtsThreshold= fragmentationThreshold= authentication= channelBandWidth= securityMode= (a.) securityMode=0 return Nothing!! (b.) securityMode=1 WEP. authenticationType= WEP. defaultTransmitKeyIndex = WEP. wepKeyLength= WEP. encryptionKeyList.Keyentry1.encryptionKey= WEP. encryptionKeyList.Keyentry2.encryptionKey= WEP. encryptionKeyList.Keyentry3.encryptionKey= WEP. encryptionKeyList.Keyentry4.encryptionKey= (c.) securityMode=2 WPA. algorithmType= WPA.sharedKey= (d.) securityMode=3 WPA. algorithmType= WPA.sharedKey= WPS.WPSMode= WPS.PINCode
Comment	
Method	GET

8.11 setIPFilter

ActionEvent: setIPFilter

Request	http://<IP>/cgi-bin/IPFilter.cgi? action=**set** permissionType= enabled= allow.enabled1= allow.startIP1= allow.endIP1=

	allow.enabled2= allow.startIP2= allow.endIP2= ……. deny.enabled1= deny.startIP1= deny.endIP1= deny.enabled2= deny.startIP2= deny.endIP2=
Response	
Comment	
Method	POST

SIPFilterSetting	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
enabled	R	Int	[0,1]	0:disabled 1:enabled
permissionType	R	Int	[0,1]	0: Deny 1: Allow

SFilterAddressEntry	Req or Opt or N	Data type	Allowed Value	notes
enabled	R	Int	[0,1]	0:disabled 1:enabled
startIP	R	Char[16]		
endIP	R	Char[16]		

8.12 getIPFilter

ActionEvent: getIPFilter

Request	http://<IP>/cgi-bin/ IPFilter.cgi? action=**get**
Response	enabled= permissionType= allow.size= allow.enabled1= allow.startIP1= allow.endIP1= allow.enabled2= allow.startIP2= allow.endIP2= ……. deny.size= deny.enabled1= deny.startIP1= deny.endIP1= deny.enabled2= deny.startIP2= deny.endIP2=
Comment	
Method	GET

9. Storage

Storage API allows applications to configure the storage devices which could be reached by the IPCAM unit.

9.1 getSDstatus

ActionEvent: getSDstatus

Request	http://<IP>/cgi-bin/sdcard.cgi?action= **getSDstatus**
Response	size= file1.name= file1.size= file1.time= file2.name= file2.size= file2.time= ……
Comment	
Method	GET

9.2 mount

ActionEvent: mount

Request	http://<IP>/cgi-bin/sdcard.cgi?action=**mount**
Response	
Comment	
Method	GET

9.3 umount

ActionEvent: umount

Request	http://<IP>/cgi-bin/sdcard.cgi?action=**umount**
Response	
Comment	
Method	GET

9.4 removeFile

ActionEvent: rm

Request	http://<IP>/cgi-bin/sdcard.cgi?action=rm&filename=
Response	
Comment	
Method	GET

10. System

System API allows applications to configure miscellaneous system settings which are not covered by any other category. These settings include Time, Syslog, etc.

// NOTE: In the future, we may switch to rsyslog instead of syslogd.

10.1 getDeviceInfo

ActionEvent: getDeviceInfo

Request	http://<IP>/cgi-bin/system.cgi?action=**get**
Response	chipVersion= sensorID= macAddress= firmwareVersion= firmwareReleasedDate= InternalName= ProductName= ModelNumber= CompanyName= Comments=
Comment	
Method	GET

10.2 setTimeSetting

ActionEvent: setTimeSetting

Request	http://<IP>/cgi-bin/time.cgi? action=**set** type=0 or ========== ====== type=1 enableDST= timezoneID= manual.year= manual.month= manual.day= manual.hour= manual.minute= manual.second= or ==========

	====== type=2 enableDST= timezoneID= ntp.ntpServerLoc1= ntp.ntpServerLoc2=
Response	
Comment	
Method	POST

STimeSetting	Req or Opt or N	Data type	Allowed Value	notes
type	R	Int	[0,2]	Table
enableDST	R	Int	[0,1]	0:disabled 1:enabled
timezoneID	R	Int	[0,24]	Table

Table: Available **type**

type	Value
TIME_CONFIG_TYPE_NONE	**0**
TIME_CONFIG_TYPE_MANUAL	**1**
TIME_CONFIG_TYPE_NTP	**2**

Table: Available **timezoneID**

timezoneID	Value
TIME_ZONE_MIN	**0**
TIME_ZONE_KWAJALEIN	**1**
TIME_ZONE_SAMOA	**2**
TIME_ZONE_HAWAII	**3**
TIME_ZONE_ALASKA	**4**
TIME_ZONE_LOS_ANGELES	**5**
TIME_ZONE_PHOENIX	**6**
TIME_ZONE_MEXICO_CITY	**7**
TIME_ZONE_NEW_YORK	**8**
TIME_ZONE_SANTIAGO	**9**
TIME_ZONE_SAO_PAULO,	**10**
TIME_ZONE_NORONHA_ISLAND,	**11**
TIME_ZONE_PRAIA,	**12**
TIME_ZONE_LONDON,	**13**
TIME_ZONE_PARIS,	**14**
TIME_ZONE_CAIRO,	**15**
TIME_ZONE_MOSCOW,	**16**
TIME_ZONE_DUBAI,	**17**
TIME_ZONE_KARACHI,	**18**
TIME_ZONE_DHAKA,	**19**
TIME_ZONE_JAKARTA,	**20**
TIME_ZONE_HONG_KONG,	**21**
TIME_ZONE_TOKYO,	**22**
TIME_ZONE_SYDNEY,	**23**
TIME_ZONE_NOUMEA,	**24**
TIME_ZONE_NewZealand,	**25**
TIME_ZONE_MAX	**26**

SNTPTimeSetting	Req or Opt or N	Data type	Allowed Value	notes
ntpServerLoc1	R	Char[128]		Only hostname
ntpServerLoc2	R	Char[128]		Only hostname

SManualTimeSetting	Req or Opt or N	Data type	Allowed Value	notes
year	R	Int	[2009,2038]	
month	R	Int	[1,12]	
day	R	Int	[1,31]	
hour	R	Int	[0,23]	

minute	R	Int	[0,59]	
second	R	Int	[0,59]	

10.3 getTimeSetting

ActionEvent: getTimeSetting

Request	http://<IP>/cgi-bin/time.cgi?action=**get**
Response	type= enableDST= timezoneID= manual.year= manual.month= manual.day= manual.hour= manual.minute= manual.second= enableDST= timezoneID= ntp.ntpServerLoc1= ntp.ntpServerLoc2=
Comment	
Method	GET

10.4 setSyslogSetting

ActionEvent: setSyslogSetting

Request	http://<IP>/cgi-bin/syslog.cgi? action=**set** localLogLevel= useRemoteLog= addressingFormatType= remoteServerAddress= remoteServerPort=
Response	
Comment	
Method	POST

SSyslogSetting	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
localLogLevel	R	Int	[0,7]	table
useRemoteLog	R	Int	[0,1]	0:disabled 1:enabled
addressingFormatTyp	R	Int	[0,1]	table

e				
remoteServerAddress	R	Char[128]		
remoteServerPort	R	Int	514,[1025,65534]	

Table: Available localLogLevel

timezoneID	Value
SLOG_EMERG	**0**
SLOG_ALERT	**1**
SLOG_CRIT	**2**
SLOG_ERR	**3**
SLOG_WARNING	**4**
SLOG_NOTICE	**5**
SLOG_INFO	**6**
SLOG_DEBUG	**7**

Table: Available addressingFormatType

addressingFormatType	Value
IP_TYPE	**0**
HOSTNAME_TYPE	**1**

10.5 getSyslogSetting

ActionEvent: getSyslogSetting

Request	http://<IP>/cgi-bin/syslog.cgi ?action=**get**
Response	localLogLevel= useRemoteLog= addressingFormatType= remoteServerAddress= remoteServerPort=
Comment	
Method	GET

10.6 getSyslogFile

ActionEvent: getSyslogFile

Request	http://<IP>/syslog.dump
Response	Content of syslog.
Comment	
Method	GET

10.7 syslogClear

ActionEvent: syslogClear

Request	http://<IP>/cgi-bin/syslog.cgi?action=clear
Response	
Comment	Clear syslog.
Method	GET

ActionEvent: getSystemStatus

Request	**http://<IP>/cgi-bin/systemStatus.cgi?action=get**
Response	
Comment	
Method	**GET**

11. Admin

Admin API enables applications to execute administrative tasks on the IPCAM unit.
The tasks include add/delete users, upgrade firmware, etc.

11.1 addUser

ActionEvent: addUser

Request	http://<IP>/cgi-bin/users.cgi? action=**add** index= username=<username> password=<password> privilege=<privilege>
Response	
Comment	
Method	POST

SUserSetting	Req or Opt or N	Data type	Allowed Value	notes
index	R	Int		Based on GUI index
username	R	Char[30]		Reject []\&|";< >*`'#$^.?/{} Length limited:[4,29]
password	R	Char[30]		Reject []<>"#'\&|;?* Length limited:[4,29]
privilege	R	Int	[0,2]	table

Table: Available privilege

privilege	Value
USER_PRIVILEGE_VIEW	**0**
USER_PRIVILEGE_ADMIN	**1**
USER_PRIVILEGE_REMOTE_VIEW	**2**

11.2 deleteUser

ActionEvent: deleteUser

Request	http://<IP>/cgi-bin/users.cgi? action=**delete** username=<username>
Response	
Comment	
Method	POST

11.3 getUsers

ActionEvent: getUsers

Request	http://<IP>/cgi-bin/users.cgi?action=**getUsers**
Response	Size= User1.index= User1.username= User1.password= User1.privilege= … User2.username= User2.password= User2.privilege=
Comment	
Method	GET

11.4 updateUser

ActionEvent: updateUser

Request	http://<IP>/cgi-bin/users.cgi? action= **update** index= username=<xxxx> password= privilege=
Response	

Comment	
Method	POST

11.5 setHTTP

ActionEvent: setHTTP

Request	http://<IP>/cgi-bin/http.cgi? action= **set** enabled= port=
Response	
Comment	
Method	POST

SHTTPSetting	Req or Opt or N	Data type	Allowed Value	notes
enabled	R	Int	[0,1]	0:disabled 1:enabled
port	R	Int	80,[1025,65534]	

11.6 setHTTP/HTTPS

ActionEvent: setHTTP/HTTPS

Request	http://<IP>/cgi-bin/http.cgi? action= **setAll** enabled= port= httpsEnabled= httpsPort=
Response	
Comment	
Method	POST

11.7 getHTTP

ActionEvent: getHTTP

Request	http://<IP>/cgi-bin/http.cgi?action= **get**
Response	enabled= port=
Comment	
Method	GET

11.8 setHTTPS

ActionEvent: setHTTPS

Request	http://<IP>/cgi-bin/https.cgi? action= **set** enabled= port=
Response	
Comment	
Method	POST

SHTTPSSetting	Req or Opt or N	Data type	Allowed Value	notes
enabled	R	Int	[0,1]	0:disabled 1:enabled
port	R	Int	443,[1025,65534]	

11.9 getHTTPS

ActionEvent: getHTTPS

Request	http://<IP>/cgi-bin/https.cgi?action= **get**
Response	enabled= port=
Comment	
Method	GET

11.10 resetToDefault

ActionEvent: resetToDefault

Request	http://<IP>/cgi-bin/reset.cgi?action= **reset**
Response	
Comment	Reset all settings to factory default
Method	GET

11.11 upgradeFirmware

ActionEvent: upgradeFirmware

Request	http://<IP>/cgi-bin/upgradeFirmware.cgi? action= **upgrade**

	Followed by the IPCam firmware
Response	
Comment	Upgrade the system firmware upon this request
Method	POST

SFWUPGRADE	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
filename	R	Char[64]		Reject ‘ “
status	R	Int		Reserved ,only get

11.12 reboot

ActionEvent: reboot

Request	http://<IP>/cgi-bin/reboot.cgi?action=**reboot**
Response	
Comment	Reboot the system
Method	GET/POST

11.13 importConfigFile

ActionEvent: importConfigFile

Request	http://<IP>/cgi-bin/ConfigFile.cgi? action= **set** filename =
Response	
Comment	
Method	POST

SConfigFile	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
filename	R	Char[64]		Reject ‘ “

11.14 exportConfigFile

ActionEvent: exportConfigFile

Request	http://<IP>/cgi-bin/ConfigFile.cgi?action=**get**
Response	
Comment	
Method	get

11.15 setPWDComplexity

ActionEvent: setPWDComplexity

Request	http://<IP>/cgi-bin/complexity.cgi? action= **set** pwdRule1Enabled = pwdRule2Enabled = pwdRule3Enabled =
Response	
Comment	
Method	POST

SComplexityPWDSetting	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
pwdRule1Enabled	R	Int	[0,1]	0:disabled 1:enabled
pwdRule2Enabled	R	Int	[0,1]	0:disabled 1:enabled
pwdRule3Enabled	R	Int	[0,1]	0:disabled 1:enabled

Note:
1. PwdRule1Enabled must contain at least three of the following forms: lower case letters, upper case letters, digits, and special characters.
2. PwdRule2Enabled cannot include any character which occurs three or more times consecutively.
3. PwdRule3Enabled cannot repeat or reverse the user name.

11.16 getPWDComplexity

ActionEvent: getPWDComplexity

Request	http://<IP>/cgi-bin/complexity.cgi?action=**get**
Response	pwdRule1Enabled = pwdRule2Enabled = pwdRule3Enabled =
Comment	
Method	GET

12. Capability

ActionEvents

ActionEvent	Description
getCapability	Get camera **Capability.**
getVideoCodecs	Get video codecs
getResolutions	Get video resolutions
getAudioCodecs	Get audiocodecs

12.1 getCapability

ActionEvent: getCapability

Request	http://<IP>/cgi-bin/**capability**.cgi?action=**get**
Response	[media] channels=2 videoCodecs=H264,MJPEG,MPEG4 audioCodesc=PCMA,PCMU, AMR resolutions=1280x800,640x400,320x192 frameRate=2,3,5,7,10,15,20,25,30 bitrate=64,128,256,384,512,768,1500,2000,4000,6000,8000,10000,12000
Comment	
Method	GET

12.2 getVideoCodecs

ActionEvent: getVideoCodecs

Request	http://<IP>/cgi-bin/**capability**.cgi?action=**getVideoCodecs**
Response	videoCodecs=H264,MJPEG,MPEG4
Comment	
Method	GET

12.3 getResolutions

ActionEvent: getResolutions

Request	http://<IP>/cgi-bin/**capability**.cgi?action=getResolutions
Response	resolutions=1280x800,640x400,320x192
Comment	
Method	GET

12.4 getAudioCodecs

ActionEvent: getAudioCodecs

Request	http://<IP>/cgi-bin/**capability**.cgi?action=**getAudioCodecs**
Response	audioCodesc=PCMA,PCMU,AMR
Comment	
Method	GET

13. Motion detection

Motion detection API allows applications to
1) set/get the motion detection setting

13.1 setMotionDetection

ActionEvent: setMotionDetection

Request	http://<IP>/cgi-bin/motiondetection.cgi? action=**set** enabled=1 channelIndex detectionInterval= region1.enabled= region1.sensitivity= region1.threshold= region1.x= region1.y= region1.x1= region1.y1= region2.enabled= region2.sensitivity= region2.threshold= region2.x= region2.y= region2.x1= region2.y1= region3.enabled= region3.sensitivity= region3.threshold= …….
Response	
Comment	
Method	POST

SMDRegionEntry	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
enabled	R	Int	[0,1]	
sensitivity	R	Int	[1, 100]	
threshold	R	Int	[1, 100]	
x	R	Int	>=0	
y	R	Int	>=0	
x1	R	Int	>=0	

y1	R	Int	>=0	

Note:
Sensitivity: When sensitivity is 90 (High value), the motion detection is easily triggered.
Threshold: When threshold is 10 (low value), the motion detection is easily triggered.

SMDEntry	Req or Opt or N	Data type	Allowed Value	notes
enabled	R	Int	[0,1]	
channelIndex	R	Int	[1, 5]	
detectionInterval	R	Int	>0	millisecond

13.2 getMotionDetection

ActionEvent: getMotionDetection

Request	http://<IP>/cgi-bin/motiondetection.cgi?action=**getMD&index=<index>**
Response	enabled=1 detectionInterval= region.size region1.enabled= region1.sensitivity= region1.threshold= region1.x= region1.y= region1.x1= region1.y1= region2.enabled= region2.sensitivity= region2.threshold= region2.x= region2.y= region2.x1= region2.y1= region3.enabled= region3.sensitivity= region3.threshold= ……
Comment	
Method	GET

13.3 getMotionDetections

ActionEvent: getMotionDetections

Request	http://<IP>/cgi-bin/ motiondetection.cgi?action=**get**
Response	size= MD1.enabled=1 MD1.channelIndex MD1.detectionInterval= MD1.region.size MD1.region1.enabled= MD1.region1.sensitivity= MD1.region1.threshold= MD1.region1.x= MD1.region1.y= MD1.region1.x1= MD1.region1.y1= MD1.region2.enabled= MD1.region2.sensitivity= MD1.region2.threshold= MD1.region2.x= MD1.region2.y= MD1.region2.x1= MD1.region2.y1= MD1.region3.enabled= MD1.region3.sensitivity= MD1.region3.threshold= MD1.region3.x= MD1.region3.y= MD1.region3.x1= MD1.region3.y1= ……
Comment	
Method	GET

14. Event

Event API allows applications to
1) set/get the event setting
2) set/get the notification setting

14.1 setEventSetting

ActionEvent: setEventSetting

Request	http://<IP>/cgi-bin/event.cgi? action= **setEventSetting** R1index= R1enabled= R1name= R1eventID= R1sched.type= R1sched.time= R1actions= R2index=… …
Response	
Comment	
Method	POST

SEventScheduleSetting	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
type	R	Int	[0,2]	**Table**
time	R	Char[128]		

Table: Available **type**

type	Value
EVENT_SCHEDULE_ALWAYS	**0**
EVENT_SCHEDULE_WEEKLY	**1**
EVENT_SCHEDULE_NEVER	**2**

SEventRuleSetting	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
Index	R	Int	[0,10]	
enabled	R	Int	[0,1]	0:enabl

				ed 1:disab led
name	R	Char[10]		
eventID	R	Int	196609	Trigger MD Event ID numbe r 196609 **Table**
actions	R	Char[128]	ftp,smtp,samb a	**Table**

Table: Available eventID

eventID	Value
MD	196608+1
DI	196608+2
PIR Sensor	196608+3
	196608+4

Table: Available **actions**

methodType	Value
ACTION_NAME_FTP	**ftp**
ACTION_NAME_EMAIL	**smtp**
ACTION_NAME_SAMBA	**samba**
ACTION_NAME_HTTP	**http**
ACTION_NAME_LIGHT	**LightLed**
ACTION_NAME_PLAYAUDIO	**playAudio**
ACTION_NAME_DO	**DO**
ACTION_NAME_RECORD_VIDEO	**RecordVideo**

14.2 addEventSetting

ActionEvent: addEventSetting

Request	http://<IP>/cgi-bin/event.cgi? action= **addEventSetting** index= enabled= name= eventID= sched.type=

	sched.time= actions=
Response	
Comment	
Method	POST

14.3 updateEventSetting

ActionEvent: updateEventSetting

Request	http://<IP>/cgi-bin/event.cgi? action= **updateEventSetting** index= enabled= name= eventID= sched.type= sched.time= actions=
Response	
Comment	
Method	POST

14.4 removeEventSetting

ActionEvent: removeEventSetting

Request	http://<IP>/cgi-bin/event.cgi? action= **removeEventSetting** index=
Response	
Comment	
Method	POST

14.5 getEventPolicy

ActionEvent: getEventPolicy

Request	http://<IP>/cgi-bin/event.cgi?action=**getEventPolicy**
Response	size= R1index= R1enabled= R1name= R1eventID= R1sched.type= R1sched.time= R1actions=

	R2index=…
Comment	
Method	GET

14.6 getEventRule

ActionEvent: getEventRule

Request	http://<IP>/cgi-bin/event.cgi?action=**getEventRule**
Response	index=0 enabled=0 name= eventID=0 sched.type=0 sched.time= actions=
Comment	
Method	GET

14.7 setEmailSetting

ActionEvent: setEmailSetting

Request	http://<IP>/cgi-bin/event.cgi? action=**setEmailSetting** senderAddress= receiverAddress1= receiverAddress2= senderName= subject= attachedVideoURLEnabled= attachedSnapShotEnabled= attachedVideoClipEnabled= authenticationMode1= port1= smtpServerHostName1 accountName1= password1= authenticationMode2= port2= smtpServerHostName2= accountName2= password2=
Response	
Comment	
Method	POST

SEmailSetting	Req or Opt or N	Data type	Allowed Value	notes
senderAddress	R	Char[64]		**E-mail format**
receiverAddress1	R	Char[64]		**E-mail format**
receiverAddress2	R	Char[64]		**E-mail format**
senderName	R	Char[64]		**Reject Table** Length limited [1,63]
subject	R	Char[64]		**Reject Table** Length limited [1,63]
attachedVideoURLEnabled	R	Int	[0,1]	0:disabled 1:enabled
attachedSnapShotEnabled	R	Int	[0,1]	0:disabled 1:enabled
attachedVideoClipEnabled	R	Int	[0,1]	0:disabled 1:enabled

Reject Table:

Parameter	Reject
senderName	[]<>"#'\&|;?*SPACE
subject	[]<>"#'\&|;?*

SMailingServer	Req or Opt or N	Data type	Allowed Value	notes
authenticationMode	R	Int		**Table**
portNo	R	Int	25,[1025,65534]	
smtpServerHostName	R	Char[64]		**Reject Table** Length limited [1,64]
accountName	R	Char[64]		**Reject Table** Length limited: [1,64]
password	R	Char[64]		**Reject Table** Length limited [8,63]

Table: Available **authenticationMode**

authenticationMode	Value
PLAIN	**0**
LOGIN	**1**
LOGIN_TLS	**2**

Reject Table:

Parameter	**Reject**
smtpServerHostName	[ ] < > " # ' \ & | ; ? *
accountName	[]\&|";< >*`'#$^.?/{}()@
TLS_ password	[]\&|:";<>?,/+=*'%# SPACE

14.8 getEmailSetting

ActionEvent: getEmailSetting

Request	http://<IP>/cgi-bin/event.cgi?action=**getEmailSetting**
Response	senderAddress= receiverAddress1= receiverAddress2= senderName= subject= attachedVideoURLEnabled= attachedSnapShotEnabled= attachedVideoClipEnabled= authenticationMode1= port1= smtpServerHostName1 accountName1= password1= authenticationMode2= port2= smtpServerHostName2= accountName2= password2=
Comment	
Method	GET

14.9 setFTPSetting

ActionEvent: setFTPSetting

Request	http://<IP>/cgi-bin/event.cgi? action= **setFTPSetting** uploadSnapShotEnabled= uploadVideoClipEnabled= addressType1= hostName1= ipAddress1= ipv6Address1= port1= accountName1= password1= passiveMode1= addressType2= hostName2= ipAddress2= ipv6Address2= port2= accountName2=

	password2= passiveMode2=
Response	
Comment	
Method	POST

SFTPSetting	Req or Opt or N	Data type	Allowed Value	notes
uploadSnapShotEnabled	R	Int	[0,1]	0:disabled 1:enabled
uploadVideoClipEnabled	R	Int	[0,1]	0:disabled 1:enabled

SFTPServer	Req or Opt or N	Data type	Allowed Value	notes
addressType	R	Int	[0,1]	Table
hostname	R	Char[64]		
ipAddress	R	Char[16]		
ipv6Address	R	Char[48]		
portNo	R	Int	21,[1025,65534]	
accountName	R	Char[64]		
password	R	Char[64]		
passiveModeEnabled	R	Int	[0,1]	

Table: Available **addressType**

AddressType	Value
IP_TYPE	**0**
HOSTNAME_TYPE	**1**

14.10 getFTPSetting

ActionEvent: getFTPSetting

Request	http://<IP>/cgi-bin/event.cgi?action= **getFTPSetting**
Response	uploadSnapShotEnabled= uploadVideoClipEnabled= addressType1= hostName1=

	ipAddress1= ipv6Address1= port1= accountName1= password1= passiveMode1= addressType2= hostName2= ipAddress2= ipv6Address2= port2= accountName2= password2= passiveMode2=
Comment	
Method	GET

14.11 setAlarmMediaInfo

ActionEvent: setAlarmMediaInfo

Request	http://<IP>/cgi-bin/event.cgi? action= **setAlarmMediaInfo** snapShotEnabled = videoClipEnabled = timeBeforeEvent= timeAfterEvent=
Response	
Comment	
Method	POST

SAlarmMediaInfo	Req or Opt or N	Data type	Allowed Value	notes
snapShotEnabled	R	Int	[0,1]	0:disabled 1:enabled
videoClipEnabled	R	Int	[0,1]	0:disabled 1:enabled
preAlarmInterval	R	Int	[1,10]	Second
postAlarmInterval	R	Int	[1,300]	Second

14.12 getAlarmMediaInfo

ActionEvent: getAlarmMediaInfo

Request	http://<IP>/cgi-bin/event.cgi?action= **getAlarmMediaInfo**
Response	snapShotEnabled = videoClipEnabled = timeBeforeEvent= timeAfterEvent=
Comment	
Method	GET

14.13 setSamba

ActionEvent: setSamba

Request	http://<IP>/cgi-bin/event.cgi? action= **setSamba** hostDns= IpAddress= Ipv6Address= UserName= Password=

	workgroup= shareDIR= addressTyep= Preserve=
Response	
Comment	
Method	POST

SambaServer	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
hostDns	R	Char[32]		
ipAddress	R	Char[16]		
ipv6Address	R	Char[48]		
userName	R	Char[16]		
password	R	Char[16]		
addressType	R	Int	[0,1]	**Table**
preserve	R	Char[12]		
workgroup	R	Char[32]		
shareDIR	R	Char[32]		

Table: Available **AddressType**

AddressType	Value
IP_TYPE	**0**
HOSTNAME_TYPE	**1**

14.14 getSamba

ActionEvent: getSamba

Request	http://<IP>/cgi-bin/event.cgi?action= **getSamba**
Response	addressType= hostDns= ipAddress= ipv6Address= userName= password= preserve= shareDIR= workGroup=
Comment	
Method	GET

14.15 setHttp

ActionEvent: setHttp

Request	http://<IP>/cgi-bin/event.cgi action= **setHttp** enabled= username= password= HttpUrl= Message=
Response	
Comment	
Method	POST

Http	Req or Opt or N	Data type	Allowed Value	notes
enabled	R	Int	[0,1]	0:disabled 1:enabled
username	R	Char[128]		
password	R	Char[128]		
HttpUrl	R	Char[16]		
Message	R	Char[16]		

14.16 getHttp

ActionEvent: getHttp

Request	http://<IP>/cgi-bin/event.cgi?action= **getHttp**
Response	Enabled= HttpUrl= Message=
Comment	
Method	GET

15. I/O Control

I/O Control API allows applications to
1) set/get the GPIO setting

15.1 setGPIOSetting

ActionEvent: setGPIOSetting

Request	http://<IP>/cgi-bin/gpio.cgi? action=set di1.port= di1.triggerType=[0,1] di1.resetIntervalAfterTriggered di1. DIenabled=0 do1.port= do1.triggerType=[2,3] do1.actionType=0 do1. triggerTime=0 do1. DOenabled=0
Response	
Comment	
Method	POST

SDIEntry	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
port	R	Int	[0,15]	**0:** Reserved **Table**
status	R	Int	[0,1]	**0:Low** **1:High**
triggerType	R	Int	[0,3]	**Table**
resetIntervalAfterTriggered	R	Int	[0,]	Reserved

SDOEntry	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
port	R	Int	[0,15]	**0:** Reserved **Table**
status	R	Int	[0,1]	**0:Low** **1:High**
triggerType	R	Int	[0,3]	**Table**
actionType	R	Int	[0,]	Reserved

Table: Available port

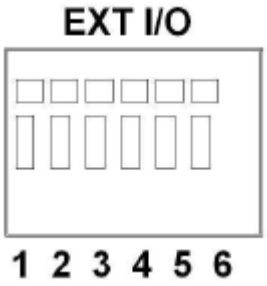
port	Value (GPIO pin)
DI	**4**
DO	**3**

Table: Available triggerType

triggerType	Value
IO_LOW	**0**
IO_HIGH	**1**
IO_OPEN	**2**
IO_GROUND	**3**
IO_RISING	**4**
IO_FALLING	**5**

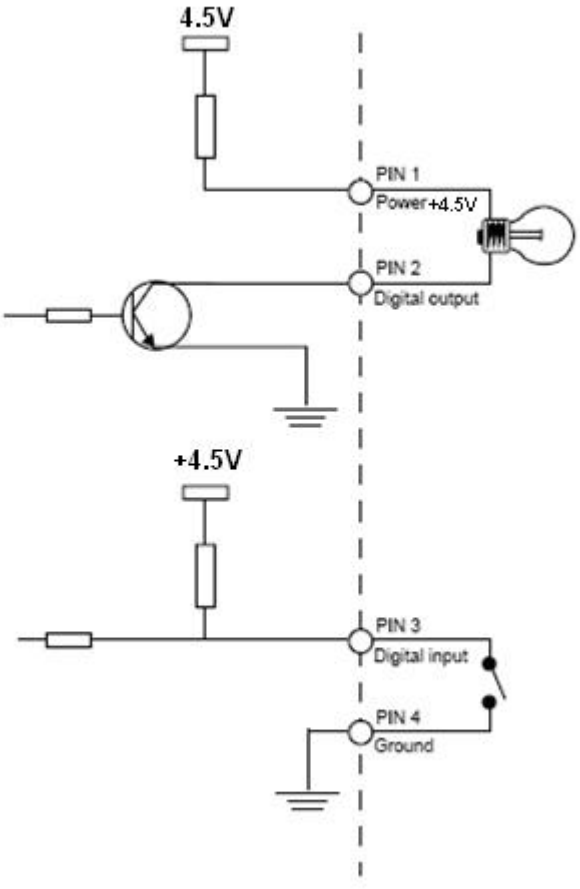
Extension I/O Terminal Block

The Network Camera provides an extension I/O terminal block to connect external input/output devices. The definition of pin is listed as below.



Pin	Function
1	Power +4.5V
2	Digital Output
3	Digital Input
4	Ground
5	RS-485 -
6	RS-485 +

DI/DO Diagram



For example: Device is ground active
1. Set DI: High DO: GROUND

	High(Open)	**Low**

DI / DO		
pin1 pin2	O/Ground	X/Open
pin1 pin2	X/ Ground	X/Open
pin2 pin4	X/Ground	X/Open
pin2 pin4	X/Ground	X/Open

2. Set DI: High DO: Open

DI / DO	High(Open)	**Low**
pin1 pin2	X/ Open	O/ Gro und
pin1 pin2	X/ Open	X/ Gro und
pin2 pin4	X/ Open	X/ Gro und
pin2 pin4	X/ Open	X/ Gro und

3. Set DI: Low DO: GROUND

DI / DO	High(Open)	**Low**
pin1 pin2	X/Open	O/Ground
pin1 pin2	X/ Open	X/ Ground
pin2 pin4	X/Open	X/Ground
pin2 pin4	X/ Open	X/ Ground

4. Set DI: Low DO: Open

DI / DO	High(Open)	**Low**
pin1	O/Ground	X/

pin2		Open
pin1 pin2	X/ Ground	X/ Open
pin2 pin4	X/Ground	X/ Open
pin2 pin4	X/ Ground	X/ Open

15.2 getGPIOStatus

ActionEvent: getGPIOStatus

Request	http://<IP>/cgi-bin/gpio.cgi?action=**get**
Response	di1.port di1.status di1.triggerType di1.resetIntervalAfterTriggered di1. Dienabled do1.port do1.status do1.triggerType do1.actionType do1. triggerTime=0 do1. Doenabled=0
Comment	
Method	GET

16. MSN

MSN API allows applications to
1) set/get the IP Camera MSNBot setting

16.1 setMSNBot

ActionEvent: setMSNBot

Request	http://<IP>/cgi-bin/msn.cgi? Action=set account= passwd= msnOpPasswd= friendlyName= buddy0.enabled= buddy0.account= buddy0.isNotifiedAcnt= buddy1.enabled= buddy1.account= buddy1.isNotifiedAcnt= buddy2.enabled= buddy2.account= buddy2.isNotifiedAcnt= buddy3.enabled= buddy3.account= buddy3.isNotifiedAcnt= buddy4.enabled= buddy4.account= buddy4.isNotifiedAcnt= webcamEnabled= alarmNotifyEnabled=
Response	
Comment	
Method	POST

MsnBuddy	Req or Opt or N	Data type	Allowed Value	notes
Enabled	R	Int	[0,1]	0:disabled 1:enabled
account	R	Char[128]		msn account
isNotifiedAcnt	R	Int	[0,1]	0:no 1:yes

Smsnbot	Req or	Data type	Allowed	notes

	Opt or N		Value	
Account	R	Char[128]		
Passwd	R	Char[128]		
msnOpPasswd	R	Char[128]		
friendlyName	R	Char[128]		
webcamEnabled	R	Int	[0,1]	0:disabled 1:enabled
alarmNotifyEnabled	R	Int	[0,1]	0:disabled 1:enabled

16.2 getMSNBot

ActionEvent: getMSNBot

Request	http://<IP>/cgi-bin/msn.cgi?action= **get**
Response	account= passwd= msnOpPasswd= friendlyName= buddy0.enabled= buddy0.account= buddy0.isNotifiedAcnt= buddy1.enabled= buddy1.account= buddy1.isNotifiedAcnt= buddy2.enabled= buddy2.account= buddy2.isNotifiedAcnt= buddy3.enabled= buddy3.account= buddy3.isNotifiedAcnt= buddy4.enabled= buddy4.account= buddy4.isNotifiedAcnt= webcamEnabled= alarmNotifyEnabled=
Comment	
Method	GET

17. PIR sensor and White LED

PIR sensor and White LED API allows applications to
1) set/get the IP Camera for PIR sensor and White LED function control.

17.1 setpirsensor

ActionEvent: set

Request	http://<IP>/cgi-bin/pirsensor.cgi? Action= **set** sensitivity = enabled=
Response	
Comment	
Method	POST

pirsensor	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
sensitivity	R	Int	[1,10]	
enabled	R	Int	[0,1]	**0:disabled** **1:enaled**

17.2 getpirsenor

ActionEvent: get

Request	http://<IP>/cgi-bin/pirsensor.cgi?action=**get**
Response	sensitivity = enabled=
Comment	
Method	GET

17.3 addWled

ActionEvent: addWled

Request	http://<IP>/cgi-bin/wled.cgi action=**add** name= method = active= inactive= duringtime=
Response	
Comment	Lightmethod: LIGHT_NONE name=xxx method =0 active=0 inactive=0 duringtime=0 //=> **reserve** //---------- Lightmethod: LIGHT_ON_OFF name=xxx method =1 active= inactive= duringtime= //=>liveView //---------- Lightmethod: LIGHT_SLIDER name=xxx method =2 active=0 inactive=0 duringtime=0 //=>liveView //---------- Lightmethod: LIGHT_PULSE

	name=xxx method =3 active= inactive= duringtime=1ms~ //=>liveView、Event
Method	POST

LightSetSetting	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
name	R	Char[16]		
method	R	Int	[0,3]	table
active	R	Int		table
inactive	R	Int		table
duringtime	R	int		**ms**

Table: Available method

method	Value
LIGHT_NONE	**0**
LIGHT_ON_OFF	**1**
LIGHT_SLIDER	**2**
LIGHT_PULSE	**3**

Table: Available active & inactive

Active inactive	Value
IMMEDIATE_OFF	**0**
IMMEDIATE_ON	**1**
FADE_TO_OF	**2**
FADE_TO_10	**10**
FADE_TO_20	**20**
FADE_TO_30	**30**
FADE_TO_40	**40**
FADE_TO_50	**50**
FADE_TO_60	**60**
FADE_TO_70	**70**
FADE_TO_80	**80**
FADE_TO_90	**90**
FADE_TO_100	**100**

17.4 deleteWled

ActionEvent: deleteWled

Request	http://<IP>/cgi-bin/wled.cgi action=**delete** name=
Response	
Comment	
Method	POST

17.5 setWled

ActionEvent: set

Request	http://<IP>/cgi-bin/wled.cgi action=**set** name= method = active= inactive= duringtime=
Response	
Comment	
Method	POST

17.6 getwledall

ActionEvent: getall

Request	http://<IP>/cgi-bin/wled.cgi?action=**getall**
Response	size= led1.name= led1.method = led1.active= led1.inactive= led1.duringtime=
Comment	
Method	GET

17.7 updateWled

ActionEvent: updateWled

Request	http://<IP>/cgi-bin/wled.cgi? Action= **update** name= method = active= inactive= duringtime=
Response	
Comment	
Method	POST

17.8 getWled

ActionEvent: get

Request	http://<IP>/cgi-bin/wled.cgi?action=**get**&name=<xxxx>
Response	name= method = active= inactive= duringtime=
Comment	
Method	GET

17.9 setLiveViewWLEDSetting

ActionEvent: set

Request	http://<IP>/cgi-bin/LVwled.cgi? Action=**set** method = active= inactive= duringtime=
Response	
Comment	
Method	POST

17.10 getLiveViewWLEDSetting

ActionEvent: get

Request	http://<IP>/cgi-bin/ Lvwled.cgi?action=**get**
Response	method = active= inactive= duringtime=
Comment	
Method	GET

a.

17.11 setLightCTL

ActionEvent: set

Request	http://<IP>/cgi-bin/wledctl.cgi? Action=**set** method= level=
Response	
Comment	
Method	POST

LightSetSetting	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
method	R	Int	[0,3]	table
level	R	Int	[1,10]	

17.12 getLightCTL

ActionEvent: get

Request	http://<IP>/cgi-bin/ wledctl.cgi?action=**get**
Response	method= level=
Comment	
Method	GET

18. PTZ

PTZ API allows applications to
1) Provide CGI commands for PTZ function control.l.

ptz	Req or Opt or N	Data type	Allowed Value	notes
move	R	char[32]	home up down left right upleft upright downleft downright	
motion	R	char[32]	request drop query	
relative	R	Int	[0,1]	**0:** absolute **1:** relative
degree	R	Int	[-360,360]	**Based on the limit of hardware**
step	R	Int	[-9999,9999]	**Based on the limit of hardware**
enabled	R	Int	[0,1]	**0:disabled autofocus** **1:autofocus**
pan	R	Int	[-100,100]	
tilt	R	Int	[-100,100]	
zoom	R	Int	[-100,100]	
focus	R	Int	[-100,100]	
imagewidth	R	Int		**depends on resolution**
imageheight	R	Int		**depends on resolution**
centerX	R	Int		**depends on resolution**
centerY	R	Int		**depends on resolution**

18.1 Direction

ActionEvent: direction

Request	http://<IP>/cgi-bin/ptz.cgi? action=direction move=left
Response	
Comment	move =home up down left right upleft upright downleft downright
Method	POST

18.2 Pan

ActionEvent: pan

Request	http://<IP>/cgi-bin/ptz.cgi? action=pan relative= degree =
Response	
Comment	relative=0 degree range -180 to 180 Pan the device relative to the (0, 0)position //===================================== relative=1 degree range -360 to 360 Pan the device n degrees relative to the current position
Method	POST

18.3 tilt

ActionEvent: tilt

Request	http://<IP>/cgi-bin/ptz.cgi? action=titl relative= degree =
Response	
Comment	relative=0 degree range -180 to 180 Pan the device relative to the (0, 0)position //===================================== relative=1 degree range -360 to 360 Tilt the device n degrees relative to the current position
Method	POST

18.4 Zoom

ActionEvent: zoom

Request	http://<IP>/cgi-bin/ptz.cgi? action=zoom relative= step =
Response	
Comment	relative=0

	step range 0 to 9999 Zoom the device n steps //========== relative=1 step range -9999 to 9999 Zoom the device n steps relative to the current position; Positive values mean zoom in, and negative values mean zoom out.
Method	POST

18.5 Focus

ActionEvent: focus

Request	http://<IP>/cgi-bin/ptz.cgi? action= **focus** enabled=0 relative= step =
Response	
Comment	enabled=0，disabled autofocus ； enabled=1，autofocus //========== relative=0，step range 0 to 9999 move focus n steps //========== relative=1，step range -9999 to 9999 Move device n steps relative to the current position; Positive values mean focus far, and negative values mean focus near.
Method	POST

18.6 Iris

ActionEvent: iris

Request	http://<IP>/cgi-bin/ptz.cgi? action= **iris** relative= step =
Response	
Comment	Relative=0 step range 0 to 9999 Move iris n steps //========== relative=1，step range -9999 to 9999 Move device n steps relative to the current position;
Method	POST

18.7 continuouspantiltmove

ActionEvent: continuouspantiltmove

Request	http://<IP>/cgi-bin/ptz.cgi? action= **continuouspantiltmove** pan= tilt=
Response	
Comment	Pan range -100 ~100 Titl range -100 ~100 Continuous pan/tilt motion. Positive values mean right (pan) and up (tilt), negative values mean left (pan) and down (tilt). “0,0” means stop.
Method	POST

18.8 continuouszoommove

ActionEvent: continuouszoommove

Request	http://<IP>/cgi-bin/ptz.cgi? action= **continuouszoommove** zoom =
Response	
Comment	zoom range -100 ~100 Continuous zoom motion. Positive values mean zoom in and negative values mean zoom out. “0” means stop.
Method	POST

18.9 continuousfocusmove

ActionEvent: continuousfocusmove

Request	http://<IP>/cgi-bin/ptz.cgi? action= **continuousfocusmove** focus =
Response	
Comment	focus range -100 ~100 Continuous focus motion. Positive values mean focus near and negative values mean focus far. “0” means stop..
Method	POST

18.10 absolutepoint

ActionEvent: absolutepoint

Request	http://<IP>/cgi-bin/ptz.cgi? action= **absolutepoint** imagewidth = imageheight= centerX= centerY=
Response	
Comment	Use this function to send the coordinates for the point in the image where the user clicked. This information is then used by the server to calculate the pan/tilt move required to (approximately) center the clicked point.
Method	POST

18.11 QueueControl

ActionEvent: Queuecontrol

Request	http://<IP>/cgi-bin/ptz.cgi? action= control motion = query
Response	
Comment	“request” requests PTZ control. “drop” drops the PTZ control or leaves the queue. “query” reports the current status for the client. For possessing clients with no peers existing in the queue, “request” will reset the control timer. For all other clients, “request” will have the same effect as “query”.
Method	POST

18.12 areazoom

ActionEvent: areazoom

Request	http://<IP>/cgi-bin/ptz.cgi? action= **areazoom** imagewidth = imageheight= centerX= centerY= zoom=
Response	
Comment	
Method	POST

19. Preset and Patrol

Preset and Patrol API allows applications to
1) set/get the IP Camera for Preset and Patrol function control.

19.1 AddPreset

ActionEvent: add

Request	http://<IP>/cgi-bin/preset.cgi? action=add name=
Response	
Comment	Associate the current position to <preset name> as a preset position in the server.
Method	POST

Preset	Req or Opt or N	Data type	Allowed Value	notes
name	R	char[65]		

19.2 deletePreset

ActionEvent: delete

Request	http://<IP>/cgi-bin/preset.cgi? action=delete name=
Response	
Comment	Remove the specified preset position associated with <preset name>.
Method	POST

19.3 gotoPreset

ActionEvent: goto

Request	http://<IP>/cgi-bin/preset.cgi? action=goto name=
Response	
Comment	Go to the specified preset position associated with

	<preset name>.
Method	POST

19.4 getPreset

ActionEvent: get

Request	http://<IP>/cgi-bin/preset.cgi?action=get
Response	size= name1= name2= ……
Comment	Get all the specified preset position associated with <preset name>.
Method	Get

19.5 sethomePreset

ActionEvent: home

Request	http://<IP>/cgi-bin/preset.cgi? action=home name=
Response	
Comment	Set the specified preset position as home preset.
Method	POST

19.6 setPatrol

ActionEvent: set

Request	http://<IP>/cgi-bin/patrol.cgi? action=set presetName1= delay1= presetName2= delay2= presetName3= delay3= presetName4= delay4= …….
Response	
Comment	
Method	POST

Patrol	Req or Opt or N	Data type	Allowed Value	notes
presetName	R	char[65]		
delay	R	Int		second

19.7 clearPatrol

ActionEvent: delete

Request	http://<IP>/cgi-bin/patrol.cgi? action=delete
Response	Use this function to clear the entire patrol setting for the device.
Comment	
Method	POST

19.8 startPatrol

ActionEvent: start

Request	http://<IP>/cgi-bin/patrol.cgi? action=start
Response	Use this function to start the patrol sequence.
Comment	
Method	POST

19.9 stopPatrol

ActionEvent: stop

Request	http://<IP>/cgi-bin/patrol.cgi? action=stop
Response	Use this function to stop the patrol sequence.
Comment	
Method	POST

19.10 getPatrol

ActionEvent: get

Request	http://<IP>/cgi-bin/patrol.cgi?action=get
Response	size= presetName1= delay1= presetName2= delay2= presetName3= delay3= presetName4= delay4= ……..
Comment	Use this function to get the patrol setting for the device.
Method	Get

20. Scheduled Event

Scheduled event allows applications to
2) set/get the IP Camera for scheduled event function control.

20.1 Add

ActionEvent: add

Request	http://<IP>/cgi-bin/schedule.cgi? action=add name= index= enabled= startHour= startMin= startSec= stopHour= stopMin= stopSec= day= scheduleAction=
Response	
Comment	
Method	POST

schedule	**Req or Opt or N**	**Data type**	**Allowed Value**	**notes**
name	R	char[65]		Unique
index	R	Int	[1,10]	0**:Reserved** , Sequence number
enabled	R	Int	[0,1]	0:disabled 1:enabled
startHour	R	Int	[0,23]	
startMin	R	Int	[0,59]	
startSec	R	Int	[0,59]	
stopHour	R	Int	[0,23]	
stopMin	R	Int	[0,59]	
stopSec	R	Int	[0,59]	
day	R	Int		table
scheduleAction	R	Int		table

Table: Available day

day	Value
Sun	**1**
Mon	**2**
Tue	**4**
Wed	**8**
Thu	**16**
Fri	**32**
Sat	**64**
Week	**128/127**

Table: Available scheduleAction

scheduleAction	Value
RECORD	**0**
REBOOT	**1**

20.2 update

ActionEvent: update

Request	http://<IP>/cgi-bin/ schedule.cgi? action=update name= index= enabled= startHour= startMin= startSec= stopHour= stopMin= stopSec= day= scheduleAction=
Response	
Comment	
Method	POST

20.3 delete

ActionEvent: delete

Request	http://<IP>/cgi-bin/ schedule.cgi? action=delete name=
Response	
Comment	
Method	POST

20.4 get

ActionEvent: delete

Request	http://<IP>/cgi-bin/schedule.cgi? action=get
Response	size= 1Enabled= 1Index= 1name= 1StartHour= 1StartMin= 1StartSec= 1StopHour=

	1StopMin= 1StopSec= 1Day= 1Action= 2Enabled= 2Index= ……
Comment	
Method	Get

20.5 getSchedule

ActionEvent: getSchedule

Request	http://<IP>/cgi-bin/schedule.cgi? action=getSchedule name=
Response	enabled= index= name= startHour= startMin= startSec= stopHour= stopMin= stopSec= day= action=
Comment	
Method	Get

21. Modification History

Revision	Date	Originator	Comments
0		Steve	Initial version 1.0
1	2009/9/9	Kenny	add API: 1.getSnapshot 2.getRtsp and setRtsp 3.getVideoCodecs getResolutions getAudioCodecs 4.getinboundChannel 5.Event Notify to HttpServer modify API: 1.getcapability 2.setAudioDevice 3.setGPIOSetting getGPIOStatus
2	2009/9/28	Kenny	1. Remove parameter level from whiteBalance Structure. 2. Add SshutterSpeedSetting and SgainSetting structure and api 3. Modify getCameraSetting and setCameraSetting
3	2009/11/12	Kenny	1. Add PTZ API and parameter 2. Check other API and parameter.
4	2009/11/17	Kenny	1. Modify IO control error. 2. Modify AppendixA InboundChanel url error
5	2009/11/18	Kenny	1. Add IO control information table.
6	2009/11/27	Kenny	1. Modify setCameraSetting getCameraSetting setEffect getEffect API

22. AppendixA InboundChanel

21.1 Introduction

Brickcom devices enable users, such as administrators, to stream live (real-time) or on-demand multimedia content from a monitor console (typically a PC or hand-held device equipped with Brickcom monitoring software) to these devices. The destination on a Brickcom device which a streaming session is intended for is called an “inbound channel”. An inbound channel is associated with one or more media peripherals, such as audio speaker, LCD display, or even file. A Brickcom device may provide zero or more inbound channels and it depends on its capability and configuration. Each channel can accept one or more external inbound streaming sessions, but only one can be played at a time. Authentication is required to prevent any inadvertent access.

21.2 Inbound Chanel Configuration

The Method of Getting Configuration

The client (the Brickcom monitoring software) uses HTTP protocol to get the inbound channel configuration on the server (the Brickcom device).
e.g. http://ip:port/cgi-bin/inboundChannel.cgi?action=get

The Syntax and Semantic of Configuration

The configuration from the HTTP server is an instance of SInboundChannelSetList.

```
typedef struct _SInboundChannelList {
    int size;
    SInboundChannelSetting channels[2];
}SInboundChannelSetList;
```

The *size* field is the number of inbound channel.
The *channels* field is an array of the inbound channel setting.

```
typedef struct _SInboundChannelSetting {
    int enabled;
    int transportType;
    int port;
    char uri[16];
    char codecType[32];
} SInboundChannelSetting;
```

The *enabled* field can enable/disable the inbound channel (This field has no effect now and the inbound channel is always enabled).
The *transportType* field determines the method of transporting stream from the monitoring software to the Brickom devices. The only supported method now is RTP over UDP (This field has no effect now and the method is always RTP over UDP).
The *port* field specifies that the RTP port for the monitoring software streams to the devices. (This field has no effect now and the RTP port is dynamically chosen by the inbound streaming server).
The *uri* field is for identifying the inbound channel. The monitoring software can use this field to initiate an inbound streaming session.
The *codecType* field specifies the codec type that the inbound channel supports. The codec types supported now are G.711 (PCMU, PCMA) and AMR. The AMR codec supported now is narrow band, 8000Hz, and 1 channel.

21.3 Session Establishment and Teardown

The monitoring software initializes a session based on the inbound channel settings. It begins via sending an INIT Request packet to the inbound streaming server which is listening on port 555. When the inbound streaming server receives the INIT Request packet, it sends back a Response packet which

contains the SDP. The SDP includes the transport ports (attribute a=*dest_port*) to which the monitoring software should stream to. From now on, the monitoring software can stream multimedia data to the server at any time until session termination. To terminate a session, the client sends a TEARDOWN Request packet, and the server sends back a Response packet to end this session. Figure 1 shows the sequence of session establishment and teardown.

The Request packet syntax:

Request = Request-Line *Message-Header CRLF [Message-Body]
Request-Line = Command SP Request-URI CRLF
Command = “INIT” | “TEARDOWN”
Request-URI = 1*CHAR

The Response packet syntax:

Response = Status-Line *Message-Header CRLF [Message-Body]
Status-Line = “OK” | “ERROR” CRLF

Message-Header = “Content-Length” “:” 1*DIGIT CRLF

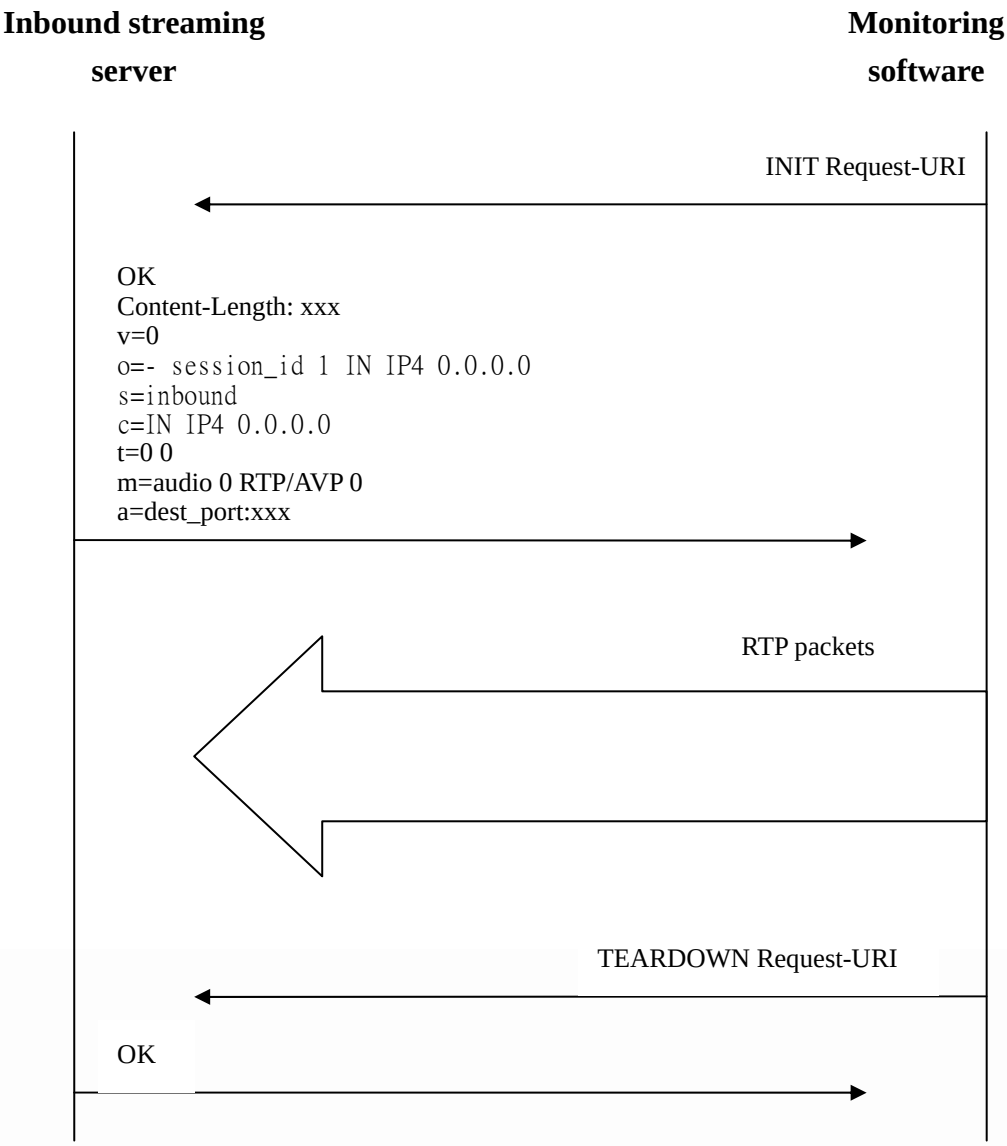


Figure 1: Session Establishment and Teardown

21.4 Authentication

[TODO]

23.AppendixB RTSP

This document specifies the external RTSP-based application programming interface of the camera and video servers. The RTSP URL is rtsp://<server ip>/channelX where <server name> is the IP address of the server. The DESCRIBE, SETUP, OPTIONS, PLAY, PAUSE and TEARDOWN methods are supported. The RTSP protocol is described in RFC 2326.