Network IP Camera Application Programming Interface (NIPCA)

Ver. 1.9.5

Document History

Version	Date	Comment
0.99a	2007-11-09	Focus on the configuration settings.
0.99b	2007-11-20	Add the Valid values. Add RS-485 commands. Refine all document.
0.99c	2007-11-21	Modify HTTP status codes, basic info, datetime DST, motion detection.
0.99d	2007-12-14	2.3: Modify HTTP status codes description. Add descriptions and examples of parameters and values. 3.3.4 3.3.5: Modify flicker, autoexposure for sensor_info.cgi sensor.cgi. 3.4.8: Modify upnpav, upnpcp for upnp.cgi. 5.1.3: Remove redundant penable, tenable, zenable for ptz_step.cgi. 5.1.5: Add p,t,z position for ptz_preset_list.cgi. 6.1.2: Add keep_alive for notify_stream.cgi.
0.99e	2007-12-26	Add the 3.5 event handling.
1.0	2008-01-14	3.1.3 Modify the request to /users/verify.cgi 3.1.6 3.1.7 Modify method and offset. Add the ID of the table dynamic DNS service providers.
1.1	2008-01-31	2.1 Fix the POST Content-Type to application/x-www-form-urlencoded. Add the ACS Stream Header.
1.2	2008-02-21	2.1 Fix the POST parameters.
1.3	2008-05-26	3.5.1 change the definition of macro block size. 3.3.4 add hue, autoexposure, autogainctrl 3.3.5 add hue, autogainctrl 3.6.8 add delaytime
1.4	2008-05-16	3.1.1 get basic information: add ‘nipca’ item 3.1.1 add ‘videoout’ item 3.3.4 add ‘inputsize’ and ‘videooutformat’ item. 3.3.5 add ‘color’ item. 3.5.1 add more actions like cifs_rec, cifs_shot… 3.5.4 add more actions and the field prerecord and postrecord. 3.5.15, 3.5.16 add keep space item. 4.1.7 add get video stream of associated profile 6.1.1, 6.1.2 add mdv# item 7.1 change url of rtsp: mpeg4 -> mp4 mjpeg -> jpeg add /live# url

1.5	2008-10-17	3.3.1, 3.3.2 add vprofileformat for video stream cgi 4.1.7 3.5.3, 3.5.4 add sw_input event 3.5.3, 3.5.4 change ‘actions’ and ‘action’ keyword to be ‘handlers’ and ‘handler’ 5.3 add an software event trigger function 4.1.5, 4.1.6 revise the part of format description 5.2.1 unify the speed range to 1-10 4.1.8 add put audio upstream (two-way audio talk) 4.1.9 add H.264 streaming cgi url Add 7.1.1-7.1.3 to support customized url entry of RTSP live stream. 3.4.1 add httpexternalport, rtspport and rtspexternalport.
1.6	2009-02-17	5.3.1 add ‘trigger’ item to indicate that client want to turn this event on or off. 3.3.1, 3.3.2, 4.1.7 Refine the definition of vprofileformat 5.1.1 add item “customizedhome” to indicate whether camera support 5.1.9 function or not. 5.1.9 add PTZ home manifest command. 3.1.13 add ‘reset sensor to default configuration’ function 3.3.5, 3.3.6 add videoinformat for some video server models 3.3.14, 3.3.15 add IR LED setting functions 3.3.16, 3.3.17 add ICR setting functions
1.7	2009-06-10	3.3.18, 3.3.19 add authentication control for live video and snapshot 3.1.1 add field ‘product’, ‘brand’ to basic information 5.1.10, 5.1.11 add Auto Patrol/Auto Pan for PTZ control function 5.1.12 Configure auot patrol preset sequence order. 3.6.5 add example html to submit firmware to ip camera. 4.1.2 add ‘profileid’ optional parameter 4.1.3 change this section to MPEG-4 elementary stream CGI. 4.1.5, 4.1.6, 4.1.9 refine the description of parameter ‘profileid’ 4.1.7 modify the format of stream with profile M-JPEG (replaced by ACVS wrapped stream) 4.1.11 add audio profile stream CGI 8.2 Add two more frame type for ACVS header. 3.3.4, 3.3.5, 3.3.6 add sharpness 3.4.14, 3.5.15 add wireless strength function and wireless site survey 3.7 SD card operations added
1.8	2009-09-30	4.1.8 Modify audio uploading method 3.7 Update whole SD card section. 3.5.11 Add enable, prefix and interval field. 3.5.13 Add enable field. 3.5.15 Add enable field.
1.9	2010-4-22	5.1.13 get, set the type of focus function: auto focus or manual focus 5.1.14 adjust the focus manually, focus in or focus out from current position Remove unsupport list. Fix wording and wrong statements.
1.9.1	2010-7-7	3.5.3. Newly add get Recorder action 3.5.4. Newly add set Recorder action 3.5.5. Newly add get Snapshot action 3.5.6. Newly add set Snapshot action 3.5.7. Newly add get Alarm out action 3.5.8. Newly add set alarm out action

1.9.2	2010-8-4	3.4.15. Newly add get HTTPS configuration 3.4.16. Newly add set HTTPS configuration
1.9.3	2010-10-31	3.3.2. 3.3.3.Modify quality not only for MJPEG and Newly add qualitymodes
1.9.4	2011-7-01	3.3.2. Newly add get sensor output configuration 3.3.3. Newly add set sensor output configuration 3.3.1. Modify the resolutions and vprofileres# description. 3.3.4. Modify the resolution description. 3.3.6. Add flicker, mirror, flip, color 5. Add /ptz/ directory to PTZ privilege group. 5.1.15., 5.1.16. Add get/set PTZ privilege group. 3.3.1. Add parameters: cur_micvol, cur_speakervol 5.3. PTDC Pan/Tilt get information 5.4. PTDC Pan/Tilt set information 5.5. PTDC Pan/Tilt other Parts 5.6. PTDC Zoom/Focus/Focus Type get information 5.7. PTDC Zoom/Focus/Focus Type set information
1.9.5	2011-11-17	3.3.4 Newly add get video type. 3.3.5 Newly add set video type. 3.3.18 Newly add set IR LED. 3.3.19 Newly add get ICR(Infrared Cut filter Removal) settings. 3.3.20 Newly add set ICR(Infrared Cut filter Removal) settings. 3.3.23 Newly add query Privacy mask information 3.3.24 Newly add get Privacy mask 3.3.25 Newly add set Privacy mask 3.4.17 Newly add list all the IP access list. 3.4.18 Newly add add, delete acces IP. 3.6.10 Newly add get Privacy mode configuration. 3.6.11 Newly add set Privacy mode configuration. 3.6.12 Newly add get TV output. 3.6.13 Newly add eet TV output. 3.6.14 Newly add get DC power. 3.6.15 Newly add set DC power. 3.6.16 Newly add get device timestamp. 3.6.17 Newly add set device timestamp. 4.1.10 Newly add get audio MS-ADPCM stream 4.1.11 Newly add get audio MU-LAW stream 4.1.12 Newly add get audio AAC stream 4.1.13 Newly add get audio A-LAW stream 5.1.17 Newly add query focus information. 5.1.18 Newly add get the current position of focus. 5.1.19 Newly add set absolutely position of focus. 5.1.20 Newly add fine-tune focus automatically. 5.1.21 Newly add PTZ direction of movement. 5.8.1 Newly add get the current digital PTZ position. 5.8.2 Newly add add, delete or goto a digital PTZ preset. 5.8.3 Newly add move digital PTZ absolutely. 5.8.4 Newly add move digital PTZ relatively. 5.8.5 Newly add digital PTZ autopan. 5.8.6 Newly add digital PTZ sequence. 3.1.1 Add field “focus” “pir” “irc” and “ir” to basic information. 3.3.1 Add “resolutionlist#”, “frameratelist#” fields. 3.3.6, 3.3.7 Add “viewwindow” fields.

		3.3.8 Add “wds” “exposuremode” , “maxshutter”, “minshutter”. “maxgain”, “noisereduction”, “wdrlevel” fields. 3.3.9, 3.3.10 Add “wds” “exposuremode” , “maxshutter”, “minshutter”. “maxgain”, “noisereduction”, “wdrenable”, “wdrlevel” fields. 3.3.11, 3.3.12 Add audio codec format value: MU-LAW, A-LAW. 3.5.1, 3.5.2 Add “percentage” “pir” fields. 3.5.4 Add “fileLenMin” field for setting recording file size by minute(s). 3.7.1 Add status values. 3.7.2 Specify response format. 6.1.1, 6.1.2 Add “irled” “autofocusbusy” field. 8.2 Add audio data format value of ACS audio header to support AAC and A-LAW audio codec format. 3.5.3, 3.5.4, 3.5.5 Errata correction. Refine TimeZone and Day Light Saving Time as below 1. TimeZone2: from “Midway Island, Samoa” to “Samoa” 2. TimeZone14: from “Bogota, Lima, Quito, Rio Branco” to “Bogota, Lima, Quito” 3. TimeZone17: from “Caracas, La Paz” to “La Paz, Georgetown” 4. TimeZone22: from “Buenos Aires, Georgetown” to “Buenos Aires” 5. TimeZone30: from “Casablanca, Monrovia, Reykjavik” to “Monrovia, Reykjavik” 6. TimeZone41: from “GMT+02:00” to “GMT+03:00” 7. TimeZone43: from “GMT+02:00” to “GMT+01:00” 8. TimeZone47: from “GMT+03:00” to “GMT+04:00” 9. TimeZone55: from “GMT+05:00 Ekaterinburg” to “GMT+06:00 Yekaterinburg” 10. TimeZone61: from “(GMT+06:00) Almaty, Novosibirsk” to “(GMT+07:00) Novosibirsk” 11. TimeZone63: from “GMT+07:00” to “GMT+08:00” 12. TimeZone67: from “GMT+08:00 Irkutsk, Ulaan Bataar” to “GMT+09:00 Irkutsk” 13. TimeZone70: from “GMT+09:00” to “GMT+10:00” 14. TimeZone80: from “(GMT+11:00) Magadan, Solomon Is., New Caledonia” to “(GMT+12:00) Magadan” 15. TimeZone81: from “Fiji, Kamchatka, Marshall Is.” to “Fiji” 16. TimeZone83: from “Nuku'alofa” to “Nukualofa” 17. TimeZone84: newly add “(GMT-04:30) Caracas” 18. TimeZone85: newly add “(GMT+11:00) Solomon Is., New Caledonia” 19. TimeZone86: newly add “(GMT) Casablanca” 20. TimeZone87: newly add “(GMT+08:00) Ulaanbaatar”

Contents

1. Overview 9

1.1. API versions 9

1.2. Valid values 9

2. HTTP Interface 10

2.1. Request messages 10

2.2. Response messages 11

2.3. Response status codes 11

3. Configuration API 12

3.1. device information 12
3.1.1. get basic information 12
3.1.2. quickly verify user 12
3.1.3. get camera info 13
3.1.4. set camera info 13
3.1.5. get system date and time 13
3.1.6. set system date and time 13

3.2. users and groups 14
3.2.1. get users 14
3.2.2. add or modify a user 14
3.2.3. delete users 14
3.2.4. get groups 14

3.3. video, sensor, audio 16
3.3.1. query stream information 16
3.3.2. get sensor output 16
3.3.3. set sensor output 17
3.3.4. get video type 17
3.3.5. set video type 17
3.3.6. get video config 17
3.3.7. set video config 18
3.3.8. sensors information 18
3.3.9. get sensors config 19
3.3.10. set sensors config 19
3.3.11. get audio config 20
3.3.12. set audio config 20
3.3.13. get microphone 20
3.3.14. set microphone 20
3.3.15. get speaker 20
3.3.16. set speaker 21
3.3.17. reset sensor to default configuration 21
3.3.18. set IR LED 21
3.3.19. get ICR(Infrared Cut filter Removal) settings 21
3.3.20. set ICR(Infrared Cut filter Removal) settings 22
3.3.21. get stream authentication setting 22
3.3.22. set stream authentication setting 22
3.3.23. query Privacy mask information 22
3.3.24. get Privacy mask 22
3.3.25. set Privacy mask 22

3.4. network *23*
3.4.1. get network config 23
3.4.2. set network config 23
3.4.3. get PPPoE 23
3.4.4. set PPPoE 24
3.4.5. get DDNS settings 24
3.4.6. set DDNS 24
3.4.7. get upnp information 24
3.4.8. set upnp information 24
3.4.9. get TCP port number for HTTP 24
3.4.10. set TCP port number for HTTP 25
3.4.11. get system wireless 25
3.4.12. set system wireless 25
3.4.13. get current wireless connection condition 26
3.4.14. do wireless site survey 26
3.4.15. get HTTPS configuration 26
3.4.16. set HTTPS configuration 26
3.4.17. list all the IP access list 27
3.4.18. add, delete acces IP 27

3.5. Motion Detection *27*
3.5.1. get motion detection 27
3.5.2. set motion detection 28
3.5.3. get Recorder action 28
3.5.4. set Recorder action 29
3.5.5. get Snapshot action 30
3.5.6. set Snapshot action 30
3.5.7. get Alarm out action 32
3.5.8. set alarm out action 32

3.6. system tools *33*
3.6.1. get digital input/output 33
3.6.2. set digital output 33
3.6.3. get LED 33
3.6.4. set LED 33
3.6.5. firmware upgrade 34
3.6.6. reboot the camera 34
3.6.7. reset all configurations to the factory default 35
3.6.8. get RS-485 settings 35
3.6.9. set RS-485 settings 36
3.6.10. get Privacy mode settings 36
3.6.11. set Privacy mode settings 36
3.6.12. get TV output 36
3.6.13. set TV output 36
3.6.14. get DC power 36
3.6.15. set DC power 37
3.6.16. get device timestamp 37
3.6.17. set device timestamp 37

3.7. SD card operation *37*
3.7.1. get information of SD card 37
3.7.2. format SD card 38
3.7.3. list items of SD card 38
3.7.4. download files of SD card 39
3.7.5. delete files of SD card 40

4. Streaming 41

4.1. Live streaming URL *41*
4.1.1. get a JPEG image 41
4.1.2. get motion JPEG video stream 41
4.1.3. get MPEG-4 elementary video stream 41
4.1.4. get MPEG-4 video stream 42
4.1.5. get audio stream 42

4.1.6. get profile video stream 42
4.1.7. put audio upstream (two-way audio talk) 43
4.1.8. get H.264 video stream 44
4.1.9. get audio WAVE stream 45
4.1.10. get audio MS-ADPCM stream 45
4.1.11. get audio MU-LAW stream 45
4.1.12. get audio AAC stream 46
4.1.13. get audio A-LAW stream 46
4.1.14. get profile audio stream 46

5. Camera Control API 47

5.1. Remote control 47
5.1.1. query PTZ information 47
5.1.2. get the current PTZ position 47
5.1.3. get the PTZ movement size in a step 47
5.1.4. set the PTZ movement size in a step 47
5.1.5. list all PTZ presets 48
5.1.6. add, delete or goto a PTZ preset 48
5.1.7. move PTZ absolutely 48
5.1.8. move PTZ relatively 48
5.1.9. get, set, goto, reset PTZ customized home position 49
5.1.10. Auto Patrol 49
5.1.11. Auto Pan 49
5.1.12. Configure sequence order of presets for Auto Patrol 50
5.1.13. get, set the type of focus function: auto focus or manual focus 50
5.1.14. adjust the focus manually, focus near or focus far from current position 50
5.1.15. get PTZ control privilege groups 51
5.1.16. set PTZ control privilege groups 51
5.1.17. query focus information 51
5.1.18. get the current position of focus 51
5.1.19. set absolutely position of focus 51
5.1.20. fine-tune focus automatically 51
5.1.21. PTZ direction of movement 52

5.2. via RS-485 52
5.2.1. do RS-485 commands 52

5.3. PTDC Pan/Tilt get information 52
5.3.1. get Pan/Tilt Position 52
5.3.2. get Pan/Tilt Position by Step 53
5.3.3. get Pan/Tilt Boundary 53
5.3.4. get Pan/Tilt Boundary by Step 53
5.3.5. get Pan/Tilt Accuracy 53
5.3.6. get Pan/Tilt Accuracy by Step 54
5.3.7. get Pan/Tilt View Angle 54
5.3.8. get Pan/Tilt View Step 54
5.3.9. get Pan/Tilt Preset Positions 54
5.3.10. get Pan/Tilt/Zoom Hardware information 55
5.3.11. get Pan/Tilt Home Position 56
5.3.12. get Pan/Tilt Patrol Speed 56
5.3.13. get Pan/Tilt Wait Time 56

5.4. PTDC Pan/Tilt set information 56
5.4.1. set Pan/Tilt Position 56
5.4.2. set Pan/Tilt Relative Position 57
5.4.3. set Pan/Tilt Position by Step 57
5.4.4. set Pan/Tilt Relative Position by Step 58
5.4.5. set Home 58
5.4.6. Restore Default Home 58
5.4.7. set Patrol Speed 59
5.4.8. set Patrol Wait Time 59

5.5. PTDC Pan/Tilt other Parts 59
5.5.1. Calibration 60

5.5.2. Single Pan 60
5.5.3. Pan Patrol 60
5.5.4. Single Patrol 60
5.5.5. User Patrol 61
5.5.6. Stop Patrol 61
5.5.7. Stop 61
5.5.8. Go Home 61
5.5.9. Goto Preset Position 62

5.6. PTDC Zoom/Focus/Focus Type get information 62
5.6.1. get Zoom Boundary 62
5.6.2. get Zoom Mag 62
5.6.3. get Zoom Boundary by Step 62
5.6.4. get Zoom Step 63
5.6.5. get Focus Boundary 63
5.6.6. get Focus Step 63
5.6.7. get Focus Type 63
5.6.8. get Zoom Accuracy 64
5.6.9. get Focus Type 64
5.6.10. get Focus Accuracy 64
5.6.11. get Home Zoom Mag 64
5.6.12. get Home Focus Step 65
5.6.13. get Home Focus Step 65
5.6.14. get Home Focus Type 65

5.7. PTDC Zoom/Focus/Focus Type set information 66
5.7.1. set Zoom Mag 66
5.7.2. set Relative Zoom Mag 66
5.7.3. set Zoom Step 66
5.7.4. set Relative Zoom Step 67
5.7.5. set Focus Step 67
5.7.6. set Relative Focus Step 68
5.7.7. set Focus Type 68

5.8. Digital PTZ Control 68
5.8.1. get the current digital PTZ position 68
5.8.2. add, delete or goto a digital PTZ preset 69
5.8.3. move digital PTZ absolutely 69
5.8.4. move digital PTZ relatively 69
5.8.5. digital PTZ autopan 69
5.8.6. digital PTZ sequence 70

6. Notification API 71

6.1. Camera status notification 71
6.1.1. get the notification status 71
6.1.2. get the notification stream 71

7. RTSP API 73

7.1. Live streaming 73
7.1.1. get URL entry of specified profile 73
7.1.2. set video config 73
7.1.3. Get live video 73

8. Appendix 75

8.1. Table used in NIPCA 75

8.2. Advanced IP-Camera Stream (ACS) Header 77

1.Overview

Network IP Camera Access Application Programming Interface (NIPCA-API) is a HTTP-based API for the networks IP camera products. Users can write program easily by calling this API to access all functionalities provided by our IP cameras including: configuration, multimedia streaming data and the control facilities.

Except Streaming, the other groups of API use the same format in transporting HTTP-based message. We will describe the command HTTP request format in the next chapter.

As for the Streaming API, the output format of streaming API depends on different IP camera model. Here we only provide a general entry point to let the IP camera outputting streaming via a permanent HTTP connection.

We also provide the RTSP interface for our IP camera.

1.1.API versions

Though we provide a common API for all IP camera models, it may not apply to some old models which were produced before the first version of this API being published. We may also publish the further version of this API in the future. So there may be some difference between different versions of API. However, all our products shall provide the API version information with every firmware version of each model.

1.2.Valid values

The following valid values are used in this document:

Values	Description
An integer	Any number between -2147483647 (-2^{31}-1) and 2147483647 (2^{31}-1).
m ... n	Any number between number m and number n.
#	A number equals or greater than 0
A string	Any string encoded by UTF-8
An IP address	A string limited to contain an IP address of the format xxx.xxx.xxx.xxx, where xxx is a number between 0 to 255. Example: 192.168.0.90
A MAC Address	A string limited to contain a MAC address of the format xx:xx:xx:xx:xx:xx, where xx is a hexadecimal value. Example: 00:40:8c:cd:00:00
A time	A string limited to contain a time of the format hh:mm:ss. Example: 23:01:14
A date	A string limited to contain a date of the format yyyy-mm-dd. Example: 2004-02-16
<value 1>, <value 2>, <value 3>, ...	Enumeration, only the given values are valid.
<*italic string*>	Every italic strings inside brackets including the brackets should be replaced by proper values.

2.HTTP Interface

An HTTP-based protocol always includes two kinds of message, request and response. IP camera prepares a service to wait and accept TCP connection request with a specified port and to process the requests message from a user defined application. In this chapter, we will describe the common format of comprising all the different request and response messages. Although our camera also can support HTTP/1.0, we recommend that a request compliant with HTTP/1.1 may encounter fewer problems. You may also refer to the RFC 2616 HTTP/1.1.

2.1.Request messages

To query information of IP camera, use the syntax

```
GET http://<camera name>/<CGI-URL>?<parameter>=<value> HTTP/1.1<CRLF>

Authorization: Basic <basic-cookie><CRLF>

Host: <camera ip-adress><CRLF>

<CRLF>
```

where,

`<CGI-URL>` is a URL of a CGI. For example, get basic information is “/common/info.cgi”.

Authorization is optional for some CGIs.

`<basic-cookie>` is the base64 encoding of userid:password. (Notes: For some models, MD5 DIGET may be used.)

`<CRLF>` is Carriage Return and Line Feed `(\r\n).`

To set values in the IP camera, you may use HTTP GET method, the syntax is

```
GET http://<camera name>/<CGI-URL>

?<parameter>=<value>[&<parameter>=<value>...] HTTP/1.1<CRLF>

Authorization: Basic <basic-cookie><CRLF>

Host: <camera ip-adress><CRLF>

<CRLF>
```

or HTTP method POST, the syntax is

```
POST http://<camera name>/<CGI-URL> HTTP/1.1<CRLF>

Authorization: Basic <basic-cookie><CRLF>

Host: <camera ip-adress><CRLF>

Content-Type: application/x-www-form-urlencoded<CRLF>

Content-Length: <body length><CRLF>

<CRLF>

<parameter>=<value>[&<parameter>=<value>]
```

where,

`<body length>` is the length of the entity body.

`<parameter>` will be described in the following chapters. Valid characters only include alphabets([A-Za-z]), digits([0-9]) and underline(_). There is no such restriction for `<value>`. The content part of the post message should be encoded with “url-encoding” function.

2.2.Response messages

While IP Camera receives request message from user, it will do the related action then output result as response message:

```
HTTP/1.1 <HTTP code> <HTTP text><CRLF>
Content-Type: text/plain<CRLF>
Content-Length: <body length><CRLF>
<CRLF>
<parameter>=<values><CRLF>
...
```

2.3.Response status codes

The API status codes are defined here.

Table 1: HTTP status codes

HTTP code	HTTP text	Description
200	OK	The request has succeeded, but an application error may occur, please refer to each CGI response.
400	Bad Request	You used invalid or unsupported parameters or values for this IP camera.
401	Unauthorized	The request requires user authentication or the authorization was refused.
404	Not Found	This API is not supported for this IP camera.
500	Internal Error	The IP camera encountered an internal error or the API can not get the correct status.
503	Service Unavailable	The IP camera is unable to handle the request due to temporary overload.

3.Configuration API

The CGIs under /config can only be accessed by administrators. Most of CGIs are one-shot commands, which only return current configurations and status of IP camera and terminated. If you need to minitor camera status for a long time, please use 6.1.2. notify_stream.cgi instead.

3.1.device information

3.1.1.get basic information

request:

GET /common/info.cgi

No authentication required.

response:

Name	Value	Description
model	A string	model name
product	A string	product name of camera
brand	A string	brand name
version	A string	version number of firmware
build	A string	firmware build number
nipca	A string	version number of NIPCA supported (e.g. 1.2, 1.4)
name	A string	camera name
location	A string	camera location
macaddr	A MAC address	the MAC address
ipaddr	An IP address	IP address of current active network interface. (Notes, this will not be a IP address of PPPoE.)
netmask	An IP address	Subnet mask
gateway	An IP address	Default router/gateway used for connecting devices attached to different networks and networks segment.
wireless	yes, no	Or omitted if the IP camera doesn’t have a wireless.
ptz	P, T, Z	Only show supported Pan or Tilt or Zoom. For example, ptz=P,T
focus	yes, no	Or omitted if the IP camera doesn’t have a focus function.
inputs	#	The number of AlarmIN.
outputs	#	The number of AlarmOUT.
speaker	yes, no	Or omitted if the IP camera doesn’t have a speaker.
Videoout	yes, no	Or omitted if the IP camera doesn’t have video out.
pir	yes, no	Or omitted if the IP camera doesn’t have PIR.
icr	yes, no	Or omitted if the IP camera doesn’t have ICR
ir	yes, no	Or omitted if the IP camera doesn’t have IR

3.1.2.quickly verify user

request:

GET /users/verify.cgi

response:

Name	Value	Description
group	A string	the group name of the specified user in the HTTP Authorization header field.

When the authorization fails, it will return HTTP/1.0 401 Unauthorized

3.1.3.get camera info

request:
GET /config/camera_info.cgi

response:

Name	Value	Description
name	A string	camera name
location	A string	camera location

3.1.4.set camera info

request:
GET/POST /config/camera_info.cgi

parameters:

see the above table.

response:
see the above table.

3.1.5.get system date and time

request:
GET /config/datetime.cgi

response:

Name	Value	Description
method	0, 1	0: disable ntpd 1: enable ntpd
timeserver	A host or IP address	NTP time server host name or IP address.
timezone	#	time zone ID, see Table Time zone
date	A date	yyyy-mm-dd
time	A time	hh:mm:ss
dstenable	no, yes	disable or enable the DST (Daylight Saving Time)
dstauto	no, yes	set DST automatically
offset	A time	The amount of time the clock should be turned back/forward (hh:mm), due to DST.
starttime		The time when DST should be enabled in the format m.w.d/hh:mm:ss day d (0 ... 6) of week w (1 ... 5) of month m (1 ... 12). d=0 is a Sunday.
stoptime		Stop time when DST should be disabled in the same format as above.

3.1.6.set system date and time

request:
GET/POST /config/datetime.cgi

parameters:

Name	Value	Description
method	0, 1, 2	0: disable ntpd 1: enable ntpd 2: manual setting, requires date and time.
timeserver	A host or IP address	NTP time server host name or IP address.
timezone	1 ... 83	time zone ID, see Table Time zone
date	A date	yyyy-mm-dd
time	A time	hh:mm:ss
dstenable	no, yes	disable or enable the DST (Daylight Saving Time)
dstauto	no, yes	set DST automatically
offset	A time	The amount of time the clock should be turned back/forward (hh:mm), due to DST.
starttime		The time when DST should be enabled in the format m.w.d/hh:mm:ss day d (0 ... 6) of week w (1 ... 5) of month m (1 ... 12).

		d=0 is a Sunday.
stoptime		Stop time when DST should be disabled in the same format as above.

response:
see the 3.1.6 table.

3.2.users and groups

3.2.1.get users

request:
GET /config/user_list.cgi

parameters:

none or

name=<username>

response:
if no request parameter

Name	Value	Description
users	#	The total number of users.
<username> ...	<group name> ...	For example, admin=admingrp It will display all user names line by line.

if request parameter is name

Name	Value	Description
group	A string	the group which this user belongs to.

3.2.2.add or modify a user

request:
GET/POST /config/user_mod.cgi

parameters:

Name	Value	Description
name	A string	user name
password	A string	base64 encoded password
group	A string	the group which this user belongs to.

response:
see the above table.

3.2.3.delete users

request:
GET/POST /config/user_del.cgi

parameters:

name =<username1>,<username2>, ...

You can delete many users at once.

response:
name=<username1>,<username2>, ...

3.2.4.get groups

request:

GET /config/group_list.cgi

parameters:

none or

name=<groupname>

response:

if no request parameter

Name	Value	Description
groups		The total number of groups.
<groupname> ...	<user1>, ...	for example, admingrp=admin,root It will display all group names and users line by line.

if request parameter is name

Name	Value	Description
user	<user1>, ...	the user names
privilege	ptz, outputs, speaker, mic, video, notify	the permissions list which this group has.

3.3.video, sensor, audio

3.3.1.query stream information

You can get supported parameter values for your IP camera. Some parameters are optional and not displayed if not supported in your IP camera.

request:

GET /config/stream_info.cgi

GET /users/stream_info.cgi (accessible by all viewers groups)

response:

Name		Value	Description
videos		MPEG4, MJPEG, H264	available video codecs list. for example, videos=MPEG4,MJPEG
audios		PCM, ADPCM, AMR, AAC	available audio codecs list. for example, audios=PCM
resolutions		<width>x<height>,...	available video resolutions list. for example, resolutions=640x480,320x240,160x120 when the current sensor output is VGA.
resolutionlist#		<width>x<height>,...	(optional) The available resolution of the current profile #. “#” is a number from 1 to the count of profiles.
vbitrates		b1, b2, b3, ...	available bitrate (kbps) list for example, vbitrates=600,800,1000
goplengths			(optional) available GOP lengths list
qualitymodes		CBR, Fixquality	available quality mode list. for example, qualitymodes = CBR, Fixquality
framerates			available frame rates list
frameratelist#			(optional) The available framerate of the current profile #. “#” is a number from 1 to the count of profiles.
qualities			available quality
asamplerates			audio sample rates (kHz) list
abitrates			audio bitrate (kbps) list
micvol			available mic volume range from v1 to v2. for example, micvol=0...100
cur_micvol		An integer	current mic volume. 0: also mean disabled.
speakervol			speaker volume range
cur_speakervol		An integer	current speaker volume. 0: also mean disabled.
vprofileformat		<Ver#>	The current version is ‘1.5’: This value indicates whether camera support /video/video.cgi or not. Please also refer to 4.1.7
vprofilenum		#	The total number of available video streams.
vprofile#		<codec name>	video profile # (# is a number from 1 to the count of profiles)
vprofileurl#			the URL for video profile #
vprofileres#		<width>x<height>	The resolution for video profile #. The resolution depends on the current sensor output.
aprofilenum		#	The total number of available audio streams.
aprofile#		<codec name>	audio profile # (# is a number from 1 to the count of profiles)
aprofileurl#			the URL for audio profile #

3.3.2.get sensor output

request:

GET /config/sensor_output.cgi

response:

Name	Value	Description
supports	<sensor output name>,<sensor output name>,.....	Available sensor output list. For example, supports=VGA,HDTV,4VGA
current	<sensor output name>	This value indicates the current sensor output. For example, current=4VGA. Changing the current sensor output result in the change of the resolutions and vprofileres# fields of stream_info.cgi and the resolution field of video.cgi.

3.3.3.set sensor output

request:

GET/POST /config/sensor_output.cgi

parameters:

current=<sensor output name>

response:

see the above table.

3.3.4.get video type

request:

GET/POST /config/video_type.cgi

response:

Name	Value	Description
profilenumber	An integer	"1" meaning that single-stream "2" meaning that dual-stream, and "3" meaning that triple-stream. If the profile has been used in the event(or recording), you can not set it.
aspectratio	String	4:3 and 16:9

3.3.5.set video type

request:

GET/POST /config/video_type.cgi

parameters:

see the above table.

response:

see the above table.

3.3.6.get video config

request:

GET /config/video.cgi

parameters:

profileid=<video profile number>

response:

Name	Value	Description
vprofileformat	Ver#	The current version is '1.5': This value indicates whether camera support /video/video.cgi or not. Please also refer to 4.1.7
profileid	#	profile number (# is a number from 1 to the count of profiles)

resolution	<width>x<height>	The resolution depends on the current sensor output. Before you set the resolution of profile, you need to get the available video resolutions list by stream_info.cgi.
viewwindow	<width>x<height>	Digital PTZ resolution.
bitrate	An integer	in kbit/s
codec	MPEG4, MJPEG, H.264	a video codec
goplength	An integer	the MPEG GOP length.
qualitymode	CBR, Fixquality	quality mode
framerate	1 ... 30	a frame rate in fps
quality	# (0-100)	available quality

3.3.7.set video config

request:

GET/POST /config/video.cgi

parameters:

see the above table.

response:

see the above table.

3.3.8.sensors information

request:

GET /config/sensor_info.cgi

response: (only supported parameters are displayed)

Name	Value	Description
brightness	b1...b2 or b1,b2,b3	available brightness range or enumeration
contrast	c1...c2 or c1,c2,c3	available contrast range or enumeration
saturation	s1...s2 or s1,s2,s3	available saturation range or enumeration
hue	h1...h2 or h1,h2,h3	available hue range or enumeration
whitebalance	auto, fixed_indoor, fixed_outdoor, fixed_fluor, disabled	available white balances list
maxexposuretime	m1...m2 or m1,m2,m3	a range of the maxexposuretime from $1/m_1$ to $1/m_2$ second or enumeration.
backlightcomp	yes, no	has backlight compensation
noisereduction	off, low, high	a list of noise reduction level.
autoexposure	yes, no	Indicate if camera support auto exposure function
autogainctrl	yes, no	Indicate if camera support auto gain control.
inputsize	<width>x<height>	Dimension of sensor size.
videooutformat	auto: auto detect. ntsc: NTSC pal: PAL pal-m:PAL M pal-n: PAN-N	For the cameras which has an analog video output connector, this field indicates the format of the video signal
sharpness	s1...s2 or s1,s2,s3	available sharpness range or enumeration.
flicker	auto, 50, 60	anti flicker. auto or 50 or 60 Hz.
mirror	off, on	disable/enable image flip horizontally
flip	off, on	disable/enable image flip vertically
color	yes, no	Select color mode or B/W mode
wds	off, on	disable/enable WDS
exposuremode	Auto, Indoor, Outdoor, Night, Moving, Low_noise, Customize1,	available exposure mode list.

	Customize2, Customize3	
maxshutter	s1...s2 or s1,s2,s3	a range of the shutter from 1/m1 to 1/m2 second or enumeration.
minshutter	s1...s2 or s1,s2,s3	a range of the shutter from 1/m1 to 1/m2 second or enumeration.
maxgain	g1...g2 or g1,g2,g3	available exposure mode gain range or enumeration.
noisereduction	off, low, high, n1...n2 or n1,n2,n3	available noise reduction range or enumeration.
wdrlevel	w1…w2 or w1,w2,w3	available wdrlevel range or enumeration.

3.3.9.get sensors config

Please call 3.3.6.sensor_info.cgi to enumerate values of related parameters.
request:

GET /config/sensor.cgi

response: (only supported parameters are displayed.)

Name	Value	Description
brightness	An integer	the brightness
contrast	An integer	the contrast
saturation	An integer	the saturation
hue	An integer	the hue
whitebalance	auto, fixed_indoor, fixed_outdoor, fixed_fluor, disabled	the white balance
flicker	auto, 50, 60	anti flicker. auto or 50 or 60 Hz.
autoexposure	yes, no	enable or disable the auto exposure
maxexposuretime	#	The divisor of maximum exposure time (1/# second). E.g. if the maximum exposure time is 1/10, then the value of this field is 10.
backlightcomp	yes, no	backlight compensation. Will make darker objects in the foreground appear clearer if the background is very bright.
noisereduction	off, low, high	noise reduction level.
mirror	off, on	disable/enable image flip horizontally
flip	off, on	disable/enable image flip vertically
autogainctrl	yes, no	enable or disable auto gain control function
color	yes, no	Select color mode or B/W mode
videoinformat	auto: auto detect. ntsc: NTSC pal: PAL pal-m:PAL M pal-n: PAN-N	For video server, the input analog video signal could be one of many video formats such as NTSC or PAL. To let video server recognize the format of video input signal, the sensor module should be configured to match the format.
sharpness	An integer	the sharpness
wds	off, on	disable/enable WDS
exposuremode	Auto, Indoor, Outdoor, Night, Moving, Low_noise, Customize1, Customize2, Customize3	the exposure mode.
maxshutter	#	the divisor of maximum shutter (1/# second). E.g. if the maximum shutter is 1/10, then the value of this field is 10.
minshutter	#	the divisor of minimum shutter (1/# second). E.g. if the minimum shutter is 1/10, then the value of this field is 10.
maxgain	An integer	the max gain.
noisereduction	off, low, high or integer	noise reduction level.
wdrenable	off, on	Disable/enable WDR.
wdrlevel	An integer	The WDR level.

3.3.10.set sensors config

request:

GET/POST /config/sensor.cgi

parameters:

see the above 2 tables to set the valid values.

response:

see the above table.

3.3.11.get audio config

request:

GET /config/audio.cgi

parameters:

profileid=<audio profile number>

response:

Name	Value	Description
profileid	#	audio profile number (# is a number from 1 to the count of profiles)
codec	PCM, ADPCM, AMR, AAC, MU-LAW, A-LAW	the audio codec
samplerate	An integer	The clock rate for the audio sampling. (in kHz)
channel	1, 2	the audio channel number.
bitrate	An integer	The output bitrate. (in kbit/s)

3.3.12.set audio config

request:

GET/POST /config/audio.cgi

parameters:

see the above table.

response:

see the above table.

3.3.13.get microphone

request:

GET /config/mic.cgi

response:

Name	Value	Description
enable	no, yes	microphone disable/enable
volume		microphone volume. please refer 3.3.1 micvol

3.3.14.set microphone

request:

see the above table.

response:

see the above table.

3.3.15.get speaker

request:

GET /config/speaker.cgi

response:

Name	Value	Description
enable	no, yes	speaker disable/enable
volume		speaker volume. please refer 3.3.1 speakervol

3.3.16.set speaker

request:

GET/POST /config/speaker.cgi

parameters:

see the above table.

response:

see the above table.

3.3.17.reset sensor to default configuration

request:

GET/POST /config/sensor_reset.cgi

parameters:

reset=go

response:

Name	Value	Description
reset	yes, fail	the result of sensor reset

3.3.18.set IR LED

request:

GET/POST /config/irled.cgi

parameters:

Name	Value	Description
mode	on, off	turn IR LED on or off

response:

see the above table.

3.3.19.get ICR(Infrared Cut filter Removal) settings

request:

GET/POST /config/icr.cgi

response:

Name	Value	Description
mode	day, night, auto, manual, schedule	indicate whether icr is day, night, auto, manual or schedule

starttime	A time	start time of schedule (in 24hr format "hh:mm", only when mode=schedule)
		for example 07:30 means 7:30 am.
		for example 19:30 means 7:30 pm.

endtime	A time	end time of schedule (in 24hr format "hh:mm", only when mode=schedule)
		for example 07:30 means 7:30 am.
		for example 19:30 means 7:30 pm.

3.3.20.set ICR(Infrared Cut filter Removal) settings

request:
see the above table.

response:
see the above table.

3.3.21.get stream authentication setting

request:
GET /config/stream_auth.cgi

response:

Name	Value	Description
livevideo	on off	indicate whether it needs authentication to get live video stream.
snapshoturl	on off	indicate whether it needs authentication to get a snapshot.

3.3.22.set stream authentication setting

request:
see the above table.

response:
see the above table.

Note: If the value of 'livevideo' is off, then the authentication for snapshot url will be turned off automatically.

3.3.23.query Privacy mask information

request:
GET /config/privacymask_info.cgi

response:

Name	Value	Description
maxnum	#	maximum of privacy mask.
maxarea	<width>x<height>	privacy mask range of the maximum space.

3.3.24.get Privacy mask

request:
GET /config/privacymask.cgi

response:

Name	Value	Description
enable#	no, yes	disable/enable privacy mask window #.
area#	A string	privacy mask window # in the format x,y,w,h x,y is the coordinate. the 0,0 means the left top position. w,h is the width and height of the window.

3.3.25.set Privacy mask

request:
GET/POST /config/privacymask.cgi

parameters:

see the above table.

response:
see the above table.

3.4.network

3.4.1.get network config

request:
GET /config/network.cgi

response:

Name	Value	Description
dhcp	off, on	disable/Enable dynamic IP address assignment
ip	An IP address	IP address of static IP setting
netmask	An IP address	subnet mask of static IP setting
gateway	An IP address	default gateway of static IP setting
dns1	An IP address	primary DNS server of static IP setting
dns2	An IP address	secondary DNS server of static IP setting
pppoe	off, on	use PPPoE
pppoeuser	A string	PPPoE user name
pppoepass	A string	PPPoE password
ddns	off, on	disable/enable dynamic DNS service
ddnsprovider		ID of the provider, see Table dynamic DNS service providers
ddnshost	A string	DDNS host name
ddnsuser	A string	DDNS user name
ddnspass	A string	DDNS password
upnp	off, on	disable/enable UPnP
httpport	1 ... 65535	TCP port number for HTTP
httpexternalport	1 ... 65535	The external port number for UPnP NAT router to map the HTTP service port of camera
rtspport	1 ... 65535	The port number of RTSP service
rtspexternalport	1 ... 65535	The external port number for UPnP NAT router to map the RTSP service port of camera

3.4.2.set network config

request:
GET/POST /config/network.cgi

parameters:

see the above table.

response:
see the above table.

3.4.3.get PPPoE

request:
GET /config/pppoe.cgi

response:

Name	Value	Description
pppoe	off, on	disable/enable PPPoE
user		PPPoE user name

pass		PPPoE password

3.4.4.set PPPoE

request:
GET/POST /config/pppoe.cgi

parameters:

see the above table.

response:
see the above table.

3.4.5.get DDNS settings

request:
GET /config/ddns.cgi

response:

Name	Value	Description
ddns	off, on	disable/enable dynamic DNS service
provider		ID of the provider, see Table dynamic DNS service providers
host		DDNS host name
user		DDNS user name
pass		DDNS password

3.4.6.set DDNS

request:
GET/POST /config/ddns.cgi

parameters:

see the above table.

response:
see the above table.

3.4.7.get upnp information

request:
GET /config/upnp.cgi

response:

Name	Value	Description
upnpav	off, on	disable/enable UPnP AV server.
upnpcp	off, on	disable/enable UPnP CP port forward

3.4.8.set upnp information

request:
GET /config/upnp.cgi

parameters:

see the above table.

response:
see the above table.

3.4.9.get TCP port number for HTTP

request:
GET /config/httpport.cgi

response:

Name	Value	Description
httpport	1 ... 65535	TCP port number for HTTP

3.4.10.set TCP port number for HTTP

request:

GET/POST /config/httpport.cgi

see the above table.

response:

see the above table.

3.4.11.get system wireless

request:

GET /config/wireless.cgi

response:

Name	Value	Description
enable	off, on	disable/enable wireless
mode	managed, ad-hoc	The type of wireless network to associate with, managed (using an access point) or ad-hoc (not using an access point).
essid	A string	ESSID
chpatterns	A string	The pattern of available wireless channels. read-only. 1111000011110000 means channel 1,2,3,4,9,10,11,12 are available.
channel	1 ... 16	wireless channel
auth	open, shared, WPA-PSK, WPA2-PSK	Authentication method. open system, shared key , WPA-PSK or WPA2-PSK
encryption	none, WEP, TKIP, AES	when auth is open: none, WEP. when auth is shared: WEP when auth is WPA-PSK or WPA2-PSK: TKIP, AES
format	hex, ASCII	only used for WEP
keylength	64, 128	WEP key length (bits)
activekey	1 ... 4	Which WEP key to use when transmitting.
key1		The keys must match the keys in the wireless access point. They could either be in hex format or in ASCII. Hex: the string must be exactly 10 hex characters for 64-bit WEP and 26 hex characters for 128-bit WEP. (Hex chars are 0123456789ABCDEF) ASCII: The string must be exactly 5 characters for 64-bit WEP and 13 characters for 128-bit WEP.
key2		
key3		
key4		
passphrase		WPA passphrase

3.4.12.set system wireless

request:

GET/POST /config/wireless.cgi

parameters:

see the above table.

response:

see the above table.

3.4.13.get current wireless connection condition

request:

GET /config/wlansignal.cgi

response:

Name	Value	Description
signal	0…100	Current wireless channel signal strength

3.4.14.do wireless site survey

request:

GET /config/wlansurvey.cgi

response: (1 site)

Name	Value	Description
ssid	A string	SSID
signal	0…100	The signal strength indicator of wireless AP.
mode	Ad-hoc infrastructure	Wireless mode
channel	1 ... 16	wireless channel
auth	open, shared, WPA-PSK, WPA2-PSK	Authentication method. open system, shared key , WPA-PSK or WPA2-PSK
encryption	none, WEP, TKIP, AES	when auth is open: none, WEP. when auth is shared: WEP when auth is WPA-PSK or WPA2-PSK: TKIP, AES

note:

Each wireless AP (access point) found has several attributes such as the above table. ‘ssid’ is the first attribute of any one wireless AP. The camera output these attributes of all found wireless AP in sequence.

3.4.15.get HTTPS configuration

request:

GET/POST /config/https.cgi

response:

Name	Value	Description
https_enable	no, yes	HTTPS function is disabled or enabled
https_only	no, yes	HTTP function is disabled or enabled(Indicate whether only HTTPS function is enabled or not)
certificate_country	A string	Country name for self-signed certificate(2 letter code)
certificate_state	A string	State or province name for self-signed certificate
certificate_locality	A string	Locality name for self-signed certificate
certificate_organization	A string	Organization name for self-signed certificate
certificate_organization_unit	A string	Organizational unit name for self-signed certificate
certificate_common_name	A string	Common name for self-signed certificate
certificate_validity	1 ... 65535	Number of days(validity) for self-signed certificate

3.4.16.set HTTPS configuration

request:

GET/POST /config/https.cgi

parameters:

see the above table.

response:

see the above table.

3.4.17.list all the IP access list

request:
GET /config/acces_list.cgi

response:

Name	Value	Description
maxallow	An integer	Maximum number of access.
allowlist	<An IP address~An IP address>,…	All allow IP range.
maxdeny	An integer	Maximum number of access.
denylist	<An IP address~An IP address>,…	All deny IP range.

3.4.18.add, delete acces IP

request:
GET /config/acces.cgi

parameters:

Name	Value	Description
type	allow, deny	Select the type of setting.
act	add, del	Select the action to perform content.
range	An IP address ~ An IP address	This parameter applies only in the "act=add".
index	An integer	This parameter applies only in the "act=del". According to this index value in the IP access list to delete the corresponding position. The minimum index value is 0.

response:
See the above table.
Set parameters (range) successful will return the IP address of the original set value, if it fails, return "empty string"(e.g. range=\r\n).Set parameters (index) successful will return index value, if it fails, return "-1"(e.g. index=-1\r\n).

3.5.Motion Detection

3.5.1.get motion detection

There are 2 possible types of motion detection dependent on your IP camera hardware.

request:
GET /config/motion.cgi

response:
macro block type:

Name	Value	Description
enable	no, yes	disable/enable motion detection
mbmask	A hex string	The macro block mask hex string of the native screen resolution which is calculated linearly from left to right then top to bottom. The size of one macro block depends on the video resolution. However, no matter the resolution of video is, the number of macro block is always 40x30. That is there is 40 block in extension of the width of video and 30 block in height.
sensitivity	0 ... 100	sensitivity
percentage	0…100	percentage
pir	no, yes	disable/enable PIR

window type:

Name	Value	Description

totalnum	#	total motion detection window numbers. read-only.
sensitivity	0 ... 100	sensitivity
percentage	0…100	percentage
enable#	no, yes	disable/enable motion detection window #
mdw#	A string	motion detection window # in the format x,y,w,h x,y is the coordinate. the 0,0 means the left top position. w,h is the width and height of the window.
pir	no, yes	disable/enable PIR

3.5.2.set motion detection

request:

GET/POST /config/motion.cgi

parameters:

see the above table.

response:

see the above table.

3.5.3.get Recorder action

request:

GET /cgi/admin/recorder.cgi

Response:
Represented by XML

```
<config>
    <record>
        <enable>0</enable>
        <profileID>0</profileID>
        <continuous>0</continuous>
        <prerecord>0</prerecord>
        <postrecord>0</postrecord>
        <keepSpace>100</keepSpace>
        <fileLenMin>1</fileLenMin>
        <recycle>0</recycle>
        <recordTo>
            <toSamba>
                <enable>0</enable>
                <anonymous>1</anonymous>
                <user></user>
                <password></password>
                <server></server>
                <shareFolder></shareFolder>
            </toSamba>
            <toUSB>
                <enable>1</enable>
            </toUSB>
        </recordTo>
        <schedule>
            <enable>0</enable>
            <profileName>Record</profileName>
            <item01>0,0,0,1,0,0</item01>
            <item02>1,0,0,2,0,0</item02>
            <item03>2,0,0,3,0,0</item03>
            <item04>3,0,0,4,0,0</item04>
            <item05>4,0,0,5,0,0</item05>
            <item06>5,0,0,6,0,0</item06>
            <item07>6,0,0,0,0,0</item07>
            <itemSize>7</itemSize>
        </schedule>
        <triggerBy>
            <byMotion>0</byMotion>
            <byIn1>0</byIn1>
```

	<byIn2>0</byIn2> </triggerBy> </record> </config>

3.5.4.set Recorder action

request:

GET/POST /cgi/admin/recorder.cgi

parameters:

Name	Value	Description
recordEnable	0 or 1	0:disable 1:Enable recording action
profileID	#	Id of recording profile.
prerecord	#	pre-recording time in second (0~15)
postrecord	#	post-recording time in second (0~15)
keepSpace	#	keep 32~999999 Mbyte for hard disk or SD card
fileLenMin	1 ... 60	Separate the recording file in 1 … 60 minute(s). For the limitation of the maximum value of each model, please reference by each model support table.
recycle	0 or 1	0: disable 1: enable, recording recycle while the CIFS disk or USB is full
toUSB	0 or 1	0: disable 1: enable, record to USB or Samba. These two parameters must be both given and enable only one of it.
toSamba		
byMotion	0 or 1	0: disable 1: enable, events to trigger recording. These parameters must be all given and could be multiple enabled.
byIn1		
byIn2		
continuous	0 or 1	0: disable 1: enable, record continuous or by schedule. These two parameters must be both given, and enable only one of it.
schedule		
anonymous	0 or 1	0: login as account 1: login as anonymous
user	A string	at most 30 characters, can accept =>!#$%()*+,-./:;=?@[\]^_`{|}~
password	A string	at most 30 characters, can accept => !#$%()*+,-./:;=?@[\]^_`{|}~
server	A string	the server ip, at most 30 characters, can accept => _-.
shareFolder	A string	at most 30 characters, can accept => _-
item01	A string [A:0…6],[B:0…23],[C:0…59],[D:0…6],[E:0…23],[F:0…59]	A: start day(0:Sunday … 6:Saturday) B: start hour C: start minute D: end day(0:Sunday … 6:Saturday) E: stop hour F: stop minute ex. 1,1,1,1,2,2 means schedule record from Monday AM:01:01 to Monday AM:02:02 ex. 1,0,0,2,0,0 mean schedule record for a whole day of Monday. ps. This item should be set relative to itemSize
item02		
item03		
item04		
item05		
item06		
item07		
itemSize	#	save the number of schedule items(0~7), ex itemSize = 3, item01~item03’s value will be saved.

response:

see the 3.5.3 table. An extera result tag will also be presented:

Result: Represented by XML	<config> <result> <code>ok</code> </result> </config>

Result code	1. ok	Success
	2. invalidParameter	Fail in cgi checking, please check setting rules
	3. updateFailed	Fail in cgi setting, please check setting rules

3.5.5.get Snapshot action

request:

GET /cgi/admin/adv_snapshot_cont.cgi

Response:
Represented by XML
	<config>
	    <mail>
	        <smtpEnable1>0</smtpEnable1>//
	        <smtpServer1></smtpServer1>
	        <smtpPort1>25</smtpPort1>
	        <smtpUser1></smtpUser1>
	        <smtpPass1></smtpPass1>
	        <receiver1></receiver1>
	        <sender1></sender1>
	    </mail>
	    <ftp>
	        <ftpEnable1>0</ftpEnable1>
	        <ftpServer1></ftpServer1>
	        <ftpPort1>21</ftpPort1>
	        <ftpUser1></ftpUser1>
	        <ftpPass1></ftpPass1>
	        <folder1></folder1>
	        <ftpInterval>1</ftpInterval>
	        <passive1>1</passive1>
	        <fixFile1>DCS-2102</fixFile1>
	    </ftp>
	    <snapshot>
	        <enable>0</enable>
	        <continuous>0</continuous>
	        <schedule>
	            <enable>0</enable>
	            <profileName>Snapshot</profileName>
	            <item01>0,0,0,1,0,0</item01>
	            <item02>1,0,0,2,0,0</item02>
	            <item03>2,0,0,3,0,0</item03>
	            <item04>3,0,0,4,0,0</item04>
	            <item05>4,0,0,5,0,0</item05>
	            <item06>5,0,0,6,0,0</item06>
	            <item07>6,0,0,0,0,0</item07>
	            <itemSize>7</itemSize>
	        </schedule>
	        <triggerBy>
	            <byMotion>0</byMotion>
	            <byIn1>0</byIn1>
	            <byIn2>0</byIn2>
	        </triggerBy>
	        <snapTo>
	            <toFtp>1</toFtp>
	            <toSmtp>1</toSmtp>
	        </snapTo>
	    </snapshot>
	</config>

3.5.6.set Snapshot action

request:

GET/POST /cgi/admin/adv_snapshot_cont.cgi

parameters:

Name	Value	Description

enable	0 or 1	0:disable 1:Enable snapshot function
toFTP	0 or 1	0: disable 1: enable. Snapshot to FTP or SMTP. These two parameters must be both given and could be multiple enabled.
toSmtp		
continuous	0 or 1	0: disable 1: enable, These two parameters must be both given, and enable only one of it. Please note only FTP support continuous snapshot.
schedule		
byMotion	0 or 1	0: disable 1: enable, events (multiselection) to trigger snapshot. These parameters must be all given.
byIn1		
byIn2		
smtpEnable1	0 or 1	0:disable 1:Enable snapshot to SMTP1
smtpServer1	A string	SMTP Server host name or IP address, at most 60 characters, can accept => _-.
smtpUser1	A string	SMTP user name, at most 30 characters, can accept => !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
smtpPass1	A string	SMTP password, at most 30 characters, can accept => !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
receiver1	A string	to which email address, at most 60 characters, can accept => !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
sender1	A string	from which email address, at most 60 characters, can accept => !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
ftpEnable1	0 or 1	0:disable 1:Enable snapshot to FTP1
ftpServer1	A string	FTP server host name or IP address, at most 60 characters, can accept => _-.
ftpPort1	#	FTP connect server port(1~65535)
ftpUser1	A string	FTP login user name, at most 30 characters, can accept => !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
ftpPass1	A string	FTP password, at most 30 characters, can accept => !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
folder1	A string	FTP init path, at most 30 characters, can not accept => ~!@#$%^&()+{}`=[];',/|*
ftpInterval	#	the time interval for continue snapshot to FTP in second.(1~86400)
passive1	0 or 1	0:disable 1:Enable FTP passive mode
fixFile1	A string	the prefix name of picture, at most 30 characters, can not accept => ~!@#$%^&()+{}`=[];',/\\|*
item01	A string [A:0…6],[B:0…23],[C:0…59],[D:0…6],[E:0…23],[F:0…59]	A: start day(0:Sunday … 6:Saturday) B: start hour C: start minute D: end day(0:Sunday … 6:Saturday) E: stop hour F: stop minute ex. 1,1,1,1,2,2 means schedule snapshot from Monday AM:01:01 to Monday AM:02:02 ex. 1,0,0,2,0,0 mean schedule snapshot for a whole day of Monday. ps. This item should be set relative to itemSize
item02		
item03		
item04		
item05		
item06		
item07		
itemSize	#	save the number of schedule items(0~7), ex itemSize = 3, item01~item03's value will be saved.

response:

see the 3.5.5 table. An extera result tag will also be presented:

Result: Represented by XML	<config> <result> <code>ok</code> </result> </config>	
Result code	1. ok	Success
	2. invalidParameter	Fail in cgi checking, please check setting rules
	3. updateFailed	Fail in cgi setting, please check setting rules

3.5.7.get Alarm out action

request:
GET /cgi/admin/adv_do.cgi

Response:
Represented by XML

```
<config>
    <digitalOutput>
        <toOut1>
            <enable>0</enable>
            <triggerBy>
                <byMotion>0</byMotion>
                <byIn1>0</byIn1>
                <byIn2>0</byIn2>
            </triggerBy>
        </toOut1>
    </digitalOutput>
</config>
```

3.5.8.set alarm out action

request:
GET/POST /cgi/admin/adv_do.cgi

parameters:

Name	Value	Description
toOut1	0 or 1	alarm to output 1
out1ByMotion	0 or 1	0: disable 1: enable, events (multiselection) to trigger alarm out1
out1ByIn1		
out1ByIn2		

response:
see the 3.5.7 table. An extera result tag will also be presented:

Result:
Represented by XML

```
<config>
    <result>
        <code>ok</code>
    </result>
</config>
```

Result code	1. ok	Success
	2. invalidParameter	Fail in cgi checking, please check setting rules
	3. updateFailed	Fail in cgi setting, please check setting rules

3.6.system tools

3.6.1.get digital input/output

This CGI is an one-shot command, which only return current status of IP camera. If you need to minitor camera status for a long time, please use 6.1.2. notify_stream.cgi instead.

request:
GET /config/io.cgi

response: (only supported inputs and outputs are displayed)

Name	Value	Description
in1	off, on	Digital input set 1
in2	off, on	Digital input set 2
in3	off, on	Digital input set 3
in4	off, on	Digital input set 4
out1	off, on	Digital output set 1
out2	off, on	Digital output set 2
out3	off, on	Digital output set 3
out4	off, on	Digital output set 4

3.6.2.set digital output

You can only set the available digital outputs, inputs are read-only.

request:
GET/POST /config/io.cgi

parameters:

out1	off, on	Digital output set 1
out2	off, on	Digital output set 2
out3	off, on	Digital output set 3
out4	off, on	Digital output set 4

response:
see the above table.

3.6.3.get LED

request:
GET /config/led.cgi

response:

Name	Value	Description
led	on, off	enable or disable the special purpose LED.

3.6.4.set LED

request:
GET/POST /config/led.cgi

parameters:

see the above table.

response:
see the above table.

3.6.5.firmware upgrade

request:

POST /config/firmwareupgrade.cgi

The file content is provided in the HTTP body according to the format given in RFC 1867. The body is created automatically by the browser if using HTML form with input type "file".

```
Example:
POST /config/firmwareupgrade.cgi HTTP/1.0\r\n
Content-Type: multipart/form-data; boundary=AsCg5y\r\n
Content-Length: <content length>\r\n
\r\n
--AsCg5y\r\n
Content-Disposition: form-data; name="update.bin"; filename="update.bin"\r\n
Content-Type: application/octet-stream\r\n
\r\n
<firmware file content>
\r\n
--AsCg5y--\r\n
```

response:

Name	Value	Description
upgrade	ok, fail	the upgrade was successful or fail

Note:
You can use web browser (e.g. Microsoft Internet Explorer or FireFox) to transport firmware to IP camera. To do this, you should write a HTML file with a form architecture to post firmware file to camera. For example:

```
<!-- saved from url=(0022)http://internet.e-mail -->
<html>

<head>
<script language="JavaScript" type="text/javascript">
function sendUpdate()
{
   var updateForm = document.updateForm;
   document.updateForm.action = "http://" + camip.value + "/config/firmwareupgrade.cgi";
   updateForm.submit();
}
</script>
</head>

<body>
Input camera ip (ex. 192.168.1.1): <input name="camip" type="text" id="camip" value=""/>
<form enctype="multipart/form-data" method="post" action="" name="updateForm">
   Choose firmware file: <input name="upload" type="file" id="upload" value=""/>
   and click
<input name="submit6" value="commit" type="button" onclick="sendUpdate()"/>
</form>
</body>

</html>
```

3.6.6.reboot the camera

request:

GET/POST /config/system_reboot.cgi

parameters:

reboot=go

response:

Name	Value	Description
reboot	yes, fail	the reboot was successful or fail

3.6.7.reset all configurations to the factory default

request:

GET/POST /config/system_reset.cgi

parameters:

reset=go

response:

Name	Value	Description
reset	yes, fail	the reset was successful or fail

3.6.8.get RS-485 settings

request:

GET /config/rs485.cgi

response:

Name	Value	description
enable	no, yes	disable/enable RS-485
proto	custom, Dyna, Lilin, Lilin2, PelcoD, PelcoP	protocol type
devid		device ID of the RS-485 slave device. Dyna: 1 ... 223 Lilin: 1 ... 64 Lilin2: 0 … 255 PelcoD: 1 ... 255 PelcoP: 1 ... 32 custom: not applicable
baudrate	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	custom baud rate
databits	7, 8	custom data bits
parity	None, Even, Odd	custom parity
stopbits	1, 2	custom stop bits
home		custom home command
up		custom up command
down		custom down command
left		custom left command
right		custom right command
stop		custom stop command
stoppattern	A string	whether to use the custom stop command for custom command 1, 2, 3, 4. 0101 means custom command 2 and 4 with stop command.
cmdname1		custom command1 name
cmdname2		custom command2 name
cmdname3		custom command3 name
cmdname4		custom command4 name
cmdstr1		custom command1 string
cmdstr2		custom command2 string
cmdstr3		custom command3 string
cmdstr4		custom command4 string
delaytime		the interval of time between two consecutive command string being executed(in millisecond, for example 300ms).

3.6.9.set RS-485 settings

request:
GET/POST /config/rs485.cgi

see the above table.

response:
see the above table.

3.6.10.get Privacy mode settings

request:
GET /config/privacy.cgi

response:

Name	Value	Description
enable	no, yes	disable/enable Privacy mode
manual	on, off	Turn on or turn off the privacy mode

3.6.11.set Privacy mode settings

request:
GET/POST /config/ privacy.cgi

see the above table.

response:
see the above table.

3.6.12.get TV output

request:
GET /config/tvoutput.cgi

response:

Name	Value	Description
out	on, off	Disable/enable this cable connection into the TV output adapter.

3.6.13.set TV output

request:
GET/POST /config/tvoutput.cgi

see the above table.

response:
see the above table.

3.6.14.get DC power

request:
GET /config/dcpower.cgi

response:

Name	Value	Description
mode	on, off, auto, schedule	Disable/enable the camera’s DC power port.

starttime	A time	start time of schedule (in 24hr format "hh:mm", only when mode=schedule)
		for example 07:30 means 7:30 am.
		for example 19:30 means 7:30 pm.

endtime	A time	start time of schedule (in 24hr format "hh:mm", only when mode=schedule)
		for example 07:30 means 7:30 am.
		for example 19:30 means 7:30 pm.

3.6.15.set DC power

request:

GET/POST /config/dcpower.cgi

see the above table.

response:

see the above table.

3.6.16.get device timestamp

request:

GET /config/timestamp.cgi

response:

Name	Value	Description
enable	on, off	Disable/enable the camera’s time stamp (label).
showtime	on, off	Disable/enable the camera’s time stamp (time).
label	A String	The time stamp on the image.

3.6.17.set device timestamp

request:

GET/POST /config/timestamp.cgi

see the above table.

response:

see the above table.

3.7.SD card operation

3.7.1.get information of SD card

This is an one-shot request, client of NIPCA could use this request to get a summary of SD card inserted in the camera. For getting up-to-date status of SD card, client should request to notify_stream.cgi (see 6.1.2), instead of pooling this cgi frequently.

request:

GET /config/sdcard.cgi

response: (only supported inputs and outputs are displayed)

Name	Value	Description
status	ready, protected, full, invalid, over_capacity, need_reinitialize, formatting, recording	Current status of SD Card
total	integer	Total size of SD Card in KBytes
used	integer	Used size of SD Card in KBytes
free	integer	Free size of SD Card in KBytes
picture	integer	How many sub-folders in picture folder
video	integer	How many sub-folders in video folder

3.7.2.format SD card

request:

GET /config/sdcard_format.cgi

parameters:

Name	Value	Description
format	go	Do format or query SD status.

response:

see the 3.7.1 table.

3.7.3.list items of SD card

request:

GET /config/sdcard_list.cgi

parameters:

Name	Value	Description
type	picture, video	Which type of files would like to see
path	string	Indicate which files of path you want to get.
page	integer	This command would list files of a page. You can indicate which list of page you would like to see.
pagesize	integer	How many files in a page. Value=5, 10, 15, 20, 50 or 100.

response:

Name	Value	Description
sd_status	ready, invalid, protected, full	Status of SD Card.
type	picture, video	Which type of files would like to see
path	string	Indicate current path to be examined
page	integer	This command would list files of a page. You can indicate which list of page you would like to see. For example: If I have 40 items (file or folder) in `path` folder and would like to list the files in page 1 where I assume the number of items in each page is 20. The request url may be: GET /config/sdcard_list.cgi?type=picture&path=/20080229/00&page=1&pagesize=20 Where: type=picture: list the items in path 'picture' path=/20080229/00: list the items which in path 'picture/20080229/00' page=1: list the items in page 1 pagesize=20: indicate that there are 20 items in each page And the server would response the fist 20 items which in path 'picture/20080229/00'
pagesize	integer	How many items (file or folder) in a page. See more details in previous parameter.
total_file	integer	Total number of these items (file or folder)
total_page	integer	Total page of these items (file or folder)
num	integer	Number of items (file or folder) in indicated page
items	string	**Attributes of listed items** <name>|<type>|<recording type>|<size or num>**:** **where:** name: name of file or folder. type: indicate this 'name' is a file or a folder, f(file) or d(folder)； recording type: d (digital input) or m (motion) or n（normal） size or num: file's size(type=f), number of files in the folder(type=d)。 **Use '|'(pipe or vertical bar) to separate each of attribute. Use ':' (column) to divide two items.** **Example:** items=20090826|d|n|100:20090826_000000.avi|f|n|512:20090825|d|n|50:…

3.7.4.download files of SD card

request:

GET /config/sdcard_download.cgi

parameters:

Name	Value	Description
type	picture, video	Which type of files would like to see
path	string	Path of file
file	string	The file name could be got in command /config/sdcard_list.cgi. The attribute type must be 'f ' (only file can be download).

response (when file is available):

```
HTTP/1.0 200 OK<CRLF>
Content-Type: application/octet-stream<CRLF>
Content-Length: <size of file><CRLF>
<CRLF>
<Binary data of file>
```

response (when file is not available for download):

Name	Value	Description
path	string	Path of file
file	string	The name of file which is wanted to be download.
result	integer	The status of download action of indicated file. 1: file is not exists 2: the ‘file’ is a folder, it must be a file.(can’t be download)

3.7.5.delete files of SD card

request:
GET /config/sdcard_delete.cgi

parameters:

Name	Value	Description
type	picture, video	Which type of files would like to be deleted
path	string	Path of file
name	<file 1>[:<file 2>: …]	File list which would be deleted. The file name could be get in command /config/sdcard_list.cgi. Use ‘:’(column) as the split character.

response:

Name	Value	Description
num	integer	Number of items (file or folder) which in ‘name’ (deleting list).
path	string	Path of file
sd_status	ready, invalid, protected	Status of SD Card. When “invalid” and “protected” is given, which mean the delete is completed failure.
items	string	**Attributes of deleted items** <name>|<type>|<status>**:** **where:** name: name of file or folder. type: indicate this ‘name’ is a file or a folder, f(file) or d(folder) or n(unknown)； status: the status of deleting action of indicated item (file or folder), <filename>. 0: item is successfully deleted 1: item does not exist 2: item is not deleted **Use ‘|’(pipe or vertical bar) to separate each of attribute. Use ‘:’ (column) to divide two items.** **Example:** items=20090826|d|0:20090826_000000.avi|n|1: 20090826_101200.avi |f|2:… where: 20090826: this item is a directory and has been deleted successfully; 20090826_000000.avi: this item is not exists; 20090826_101200.avi: this item is a file and has not been deleted.

4.Streaming

4.1.Live streaming URL

4.1.1.get a JPEG image

Returns a JPEG image with the default resolution and compression as defined in the configuration.

request:

GET /image/jpeg.cgi

response:

```
HTTP/1.0 200 OK\r\n
Content-Type: image/jpeg\r\n
Content-Length: <image size>\r\n
\r\n
<JPEG image data>\r\n
```

4.1.2.get motion JPEG video stream

```
   Returns a multipart image stream with the default resolution and compression as defined
   in the configuration. The content type is "multipart/x-mixed-replace" and  each image
   ends with a boundary string <boundary>.
```

```
request:
   GET /video/mjpg.cgi
```

parameters:

Name	Value	Description
profileid	#	optional. If omitted, the url will output one of stream profile that match the assign format (motion JPEG).

```
response:
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=<boundary>\r\n
\r\n
--<boundary>\r\n
Content-Type: image/jpeg\r\n
Content-Length: <image size>\r\n
\r\n
<JPEG image data>\r\n
--<boundary>\r\n
Content-Type: image/jpeg\r\n
Content-Length: <image size>\r\n
\r\n
<JPEG image data>\r\n
--<boundary>\r\n
```

4.1.3.get MPEG-4 elementary video stream

```
   Returns a MPEG-4 elementary stream with assigned profile id defined in the configuration.
   The content type is "video/MP4V-ES" please refer to INAN MIME Media Types.
```

```
request:
   GET /video/MP4V-ES.cgi
```

parameters:

Name	Value	Description
profileid	#	optional. If omitted, the url will output one of stream profile that match the assign format (MPEG4 elementary stream).

```
response:
HTTP/1.0 200 OK\r\n
Content-Type: video/MP4V-ES\r\n
\r\n
<MPEG-4 ISO/IEC 14496-2 elementary stream>
```

4.1.4.get MPEG-4 video stream

Return the MPEG-4 video stream. The video data header please refer to the ACS Stream Header.

request:
GET /video/ACVS.cgi

parameters:

Name	Value	Description
profileid	#	optional. If omitted, the url will output one of stream profile that match the assign format (ACVS/MPEG4 stream).

response:

```
HTTP/1.0 200 OK\r\n
Content-type: video/ACVS\r\n
\r\n
<ACAS Video Stream Data>

Where <ACAS Video Stream Data> is defined as below:
<ACS_VideoHeader>
<MPEG4 Raw Data>
<ACS_AudioHeader>
<MPEG4 Raw Data>
...
```

<ACS_VideoHeader> is defined in **8.2 Advanced ip-Camera Stream(ACS) Header**.
<MPEG4 Raw Data> is raw data of MPEG4 video stream.

4.1.5.get audio stream

The audio data header please refer to the ACS Stream Header.

request:
GET /audio/ACAS.cgi

parameters:

Name	Value	Description
profileid	#	optional. If omitted, the url will output one of stream profile that match the assign format (ACAS/PCM stream).

response:

```
HTTP/1.0 200 OK\r\n
Content-type: audio/ACAS\r\n
\r\n
<ACAS Audio Stream Data>
```

Where <ACAS Audio Stream Data> is defined as below:
<ACS_AudioHeader>
<Audio Raw Data>
<ACS_AudioHeader>
<Audio Raw Data>
...

<ACS_AudioHeader> is defined in **8.2 Advanced ip-Camera Stream(ACS) Header**.
<Audio Raw Data> is raw data of audio stream. The format of this data depends on <ACS_AudioHeader>.

4.1.6.get profile video stream

Return the video stream associated with a specific profile. The video stream format depends on the compression type of video in that profile. Please read note below.

request:

GET /video/video.cgi

parameters:

Name	Value	Description
profileid	# (1 to the count of video profile)	If omitted, the url will output the stream of default profile (profile id = 1).

response:

```
HTTP/1.0 200 OK\r\n
Content-type: video/ACVS\r\n
\r\n
<video streaming data>
```

note:

`<video streaming data>`:

If the compression type of the designated profile is motion-JPEG(MJPEG), the stream format is different with as multipart format stream. The format of video stream with profile motion-JPEG is wrapped by ACVS header.

On the other hand, if the compression type is MPEG4 (or H.264 or other advanced compression methods), the output format follows the standard of ACVS format (See Appendix 8.2).

This url is only available while the value of item 'vprofileformat' in 3.3.1 or 3.3.2 is equal to or greater than 1.0

4.1.7.put audio upstream (two-way audio talk)

There are two requests to use this service. One of the requests is "verification request", the other is called "uploading request".

While uploading audio data from client to camera server, the client may run into some situations instead of successfully keeping uploading audio data. For example, if another client has been uploading audio data, server will disconnect the connection after client starting uploading audio. On the other hand, if client send command with wrong authentication information, the server will also reject the request from client. So a client should use verification request to test if it has the correct authentication information before uploading audio stream. In other word, if a user has past the verification request but it still got disconnecting after uploading request because of the other connect existing.

Verification request:

Request object:

```
HEAD /dev/speaker.cgi?client=<MAC address of client side>
```

Request header:

```
Authorization: Basic <base64 encode(username:password)>\r\n
Content-Type: audio/ACAS\r\n
Content-length: 0\r\n
\r\n
```

Response of verification request:

If the authorization is verified, the camera should return 200 OK to indicate client side can keep uploading request:

```
HTTP/1.0 200 OK\r\n
```

If the authorization is failed, the camera would return HTTP error code to indicate client side should stop the uploading request, for example:

```
HTTP/1.0 401 Unauthorized\r\n
```

Uploading request:

Request object:

```
POST /dev/speaker.cgi?client=<MAC address of client side>
```

Request header:

```
Authorization: Basic <base64 encode(username:password)>\r\n
Content-Type: audio/ACAS\r\n
Content-length: 4\r\n
Connection: Keep-Alive\r\n
\r\n
```

Request body:

```
<Random 4CC>

 (wait for 2 sec.)
```

```
<AAH>
<1K audio data>
<AAH>
<1K audio data>
<AAH>
<1K audio data>
...
```

Where:

`<Random 4CC>: 4-byte random char code.`

`(wait for 2 sec.): After 4-bytes random char code, client should wait for 1 sec before sending more audio data.`

```
<AAH>: the header of AAH defined as follow:
typedef struct _ACS_AudioHeader
{
unsigned long ulHdrID;                //Header ID  = 0xF6010000
unsigned long ulHdrLength;          // sizeof(ACS_AudioHeader)
unsigned long ulDataLength;         // audio data length
unsigned long ulSequenceNumber; // sequence number
unsigned long ulTimeSec;             //sample time stamp
unsigned long ulTimeUSec;           // sample time stamp
unsigned long ulDataCheckSum;    // not used…
unsigned short usFormat;              //  0x00000010  S16 LE
unsigned short usChannels;           // 1 channel
unsigned short usSampleRate;       // 8000   hz
unsigned short usSampleBits;        // 16   bits
unsigned long ulReserved;           //
}ACS_AudioHeader, *PACS_AudioHeader;
```

`<1K audio data>: audio data acquired by client side in the format specified by <AAH> header`

Response of uploading request:

There are no response for this request.

4.1.8.get H.264 video stream

Return the H.264 video stream. The video data header please refer to the ACS Stream Header.

request:

GET /video/ACVS-H264.cgi

parameters:

Name	Value	Description
profileid	#	optional. If omitted, the url will output one of stream profile that match the assign format (ACVS/H.264 stream).

response:

```
HTTP/1.0 200 OK\r\n
Content-type: video/ACVS\r\n
\r\n
<ACAS Video Stream Data>

Where <ACAS Video Stream Data> is defined as below:
<ACS_VideoHeader>
<H.264 Raw Data>
<ACS_AudioHeader>
<H.264 Raw Data>
...
```

`<ACS_VideoHeader> is defined in` **8.2 Advanced ip-Camera Stream(ACS) Header.**
`<MPEG4 Raw Data> is raw data of MPEG4 video stream.`

4.1.9.get audio WAVE stream

Return the audio stream in WAV format.

request:
GET /audio/x-wav.cgi

parameters:

Name	Value	Description
sec	#	Duration of audio streaming. 0: (default) indicate maximum duration 1-120000: duration in second

response:

```
HTTP/1.0 200 OK\r\n
Content-type: audio/x-wav\r\n
\r\n
<wave format data>

Where <wave format data> is a standard Microsoft wave file format. Please refer to MIME:
audio/x-wav.
```

4.1.10.get audio MS-ADPCM stream

Return the audio stream in MS-ADPCM format.

request:
GET /audio/ACAS-MSADPCM.cgi

response:

```
HTTP/1.0 200 OK\r\n
Content-type: audio/ACAS\r\n
\r\n
<ACAS Audio Stream Data>
```

Where <ACAS Audio Stream Data> is defined as below:
<ACS_AudioHeader>
<Audio Raw Data>
<ACS_AudioHeader>
<Audio Raw Data>
...

<ACS_AudioHeader> is defined in **8.2 Advanced ip-Camera Stream(ACS) Header**.
<Audio Raw Data> is raw data of audio stream. The format of this data depends on <ACS_AudioHeader>.

4.1.11.get audio MU-LAW stream

Return the audio stream in MU-LAW format.

request:
GET /audio/ACAS- ULAW.cgi

response:

```
HTTP/1.0 200 OK\r\n
Content-type: audio/ACAS\r\n
\r\n
<ACAS Audio Stream Data>
```

Where <ACAS Audio Stream Data> is defined as below:
<ACS_AudioHeader>
<Audio Raw Data>
<ACS_AudioHeader>
<Audio Raw Data>
...

<ACS_AudioHeader> is defined in **8.2 Advanced ip-Camera Stream(ACS) Header**.
<Audio Raw Data> is raw data of audio stream. The format of this data depends on <ACS_AudioHeader>.

4.1.12.get audio AAC stream

Return the audio stream in AAC format.

request:
GET /audio/ACAS- AAC.cgi

response:

```
HTTP/1.0 200 OK\r\n
Content-type: audio/ACAS\r\n
\r\n
<ACAS Audio Stream Data>
```

Where <ACAS Audio Stream Data> is defined as below:
<ACS_AudioHeader>
<Audio Raw Data>
<ACS_AudioHeader>
<Audio Raw Data>
...

<ACS_AudioHeader> is defined in **8.2 Advanced ip-Camera Stream(ACS) Header**.
<Audio Raw Data> is raw data of audio stream. The format of this data depends on <ACS_AudioHeader>.

4.1.13.get audio A-LAW stream

Return the audio stream in A-LAW format.

request:
GET /audio/ACAS- ALAW.cgi

response:

```
HTTP/1.0 200 OK\r\n
Content-type: audio/ACAS\r\n
\r\n
<ACAS Audio Stream Data>
```

Where <ACAS Audio Stream Data> is defined as below:
<ACS_AudioHeader>
<Audio Raw Data>
<ACS_AudioHeader>
<Audio Raw Data>
...

<ACS_AudioHeader> is defined in **8.2 Advanced ip-Camera Stream(ACS) Header**.
<Audio Raw Data> is raw data of audio stream. The format of this data depends on <ACS_AudioHeader>.

4.1.14.get profile audio stream

Return the audio stream associated with a specific profile. The audio stream format depends on the compression type of audio in that profile. Please read note below.

request:
GET /audio/audio.cgi

parameters:

Name	Value	Description
profileid	# (1 to the count of audio profile)	If omitted, the url will output the stream of default profile (profile id = 1).

response:

```
HTTP/1.0 200 OK\r\n
Content-type: audio/ACAS\r\n
\r\n
<audio streaming data>
```

note:
`<audio streaming data>`:
The audio stream is wrapped by ACAS header.

5.Camera Control API

5.1.Remote control

5.1.1.query PTZ information

request:

GET /config/ptz_info.cgi
GET /ptz/ptz_info.cgi (accessible by PTZ privilege group, see 5.1.15)

response: (only supported parameters are displayed.)

Name	Value	Description
pmax	An integer	maximum position of pan in degree
pmin	An integer	minimum position of pan in degree
tmax	An integer	maximum position of tilt in degree
tmin	An integer	minimum position of tilt in degree
zmax	An integer	maximum position of zoom
zmin	An integer	minimum position of zoom
customizedhome	no, yes	to indicate whether camera can use “customized home” function. Please refer to section 5.1.9

5.1.2.get the current PTZ position

request:

GET /config/ptz_pos.cgi

GET /ptz/ptz_pos.cgi (accessible by PTZ privilege group, see 5.1.15)

response: (only supported parameters are displayed.)

Name	Value	Description
p	An integer	the pan position
t	An integer	the tilt position
z	An integer	the zoom position

5.1.3.get the PTZ movement size in a step

request:

GET /config/ptz_step.cgi

GET /ptz/ptz_step.cgi (accessible by PTZ privilege group, see 5.1.15)

response: (only supported parameters are displayed.)

Name	Value	Description
pstep	An integer	pan movement size in a step
tstep	An integer	tilt movement size in a step
zstep	An integer	zoom movement size in a step

5.1.4.set the PTZ movement size in a step

You can specify any of the parameters you want to set.

request:

GET/POST /config/ptz_step.cgi

parameters:

see the above table.

response:

see the above table.

5.1.5.list all PTZ presets

request:

GET /config/ptz_preset_list.cgi

GET /ptz/ptz_preset_list.cgi (accessible by PTZ privilege group, see 5.1.15)

response:

Name	Value	Description
presets	<preset name1>, ...	all presets

<preset name1>
...
	<p>,<t>,<z>
	...
		the position of the preset name line by line.
		for example,
		door1=100,0
		gate1=-20,-100

5.1.6.add, delete or goto a PTZ preset

request:

GET/POST /config/ptz_preset.cgi

GET/POST /ptz/ptz_preset.cgi (accessible by PTZ privilege group, see 5.1.15)

parameters:

Name	Value	Description
name		preset name
act	add del go	add the current position to the preset delete the preset (only available at /config/ dir) go to the preset

response:

see the above table.

5.1.7.move PTZ absolutely

request:

GET/POST /config/ptz_move.cgi

GET/POST /ptz/ptz_move.cgi (accessible by PTZ privilege group, see 5.1.15)

parameters:

Name	Value	Description
p	An integer	Pans the device relative to the (0,0,0) position.
t	An integer	Tilts the device relative to the (0,0,0) position.
z	An integer	Zooms the device relative to the (0,0,0) position.

response:

see the above table. If the movement is out of boundary, you will get the actual absolute position.

5.1.8.move PTZ relatively

request:

GET/POST /config/ptz_move_rel.cgi

GET/POST /ptz/ptz_move_rel.cgi (accessible by PTZ privilege group, see 5.1.15)

parameters:

Name	Value	Description
p	-32 ... 32	Pans the device relative to the current position.
t	-32 ... 32	Tilts the device relative to the current position.

z	-32 ... 32	Zooms the device relative to the current position.

response:
see the above table. If the movement is out of boundary, you will get the actual relative p, t, z values it moves.

5.1.9.get, set, goto, reset PTZ customized home position

request:
GET/POST /config/ptz_home.cgi

GET/POST /ptz/ptz_home.cgi (accessible by PTZ privilege group, see 5.1.15)

parameters:

Name	Value	Description
act	get set go reset	get current home position. This is the default value. set current home position (only available at /config/ dir) go to home position reset home position to factory setting (only available at /config/ dir)
p	An integer	(only for act=set)Pans the device relative to the default (0,0,0) position.
t	An integer	(only for act=set)Tilts the device relative to the default (0,0,0) position.
z	An integer	(only for act=set)Zooms the device relative to the default (0,0,0) position.

response:
Return current home position.

Name	Value	Description
p	An integer	the pan position
t	An integer	the tilt position
z	An integer	the zoom position

note:
If no any parameter is given, the effect will equivalent giving ‘act=get’. If any of parameters p,t,z is given and the value of parameter ‘act’ is not ‘set’, it will be ignored by camera.

5.1.10.Auto Patrol

request:
GET/POST /config/auto_patrol.cgi

GET/POST /ptz/auto_patrol.cgi (accessible by PTZ privilege group, see 5.1.15)

parameters:

Name	Value	Description
act	go [continue] [stop]	Run PTZ’s auto patrol. act=go means run auto patrol function once act=[continue] or act=[stop] means begin continuous patrol mode and stop patrol

response:
see the above table.

note:
The item enclosed by [ ] means optional value. That optional value can be used only in some special models.

5.1.11.Auto Pan

request:
GET/POST /config/auto_pan.cgi

GET/POST /ptz/auto_pan.cgi (accessible by PTZ privilege group, see 5.1.15)

parameters:

Name	Value	Description

act	go [continue] [stop]	Run PTZ's auto pan. act=go means run auto pan function once act=[continue] or act=[stop] means begin continuous pan mode and stop pan

response:

see the above table.

note:

The item enclosed by [ ] means optional value. That optional value can be used only in some special models.

5.1.12.Configure sequence order of presets for Auto Patrol

request:

GET /config/config_auto_patrol.cgi

parameters:

Name	Value	Description
presets	<preset name1>, <preset name2>,...	A sequence of presets. The CGI auto_patrol function move camera PTZ preset by this sequence. Maximum count of preset in this sequence is 20. Note: if this parameter is not given, the camera will list current sequence.

response:

see the above table. If the count of given preset point is greater than 20, only first 20 preset in the sequence will kept by camera.

5.1.13.get, set the type of focus function: auto focus or manual focus

request:

GET/POST /config/focus_type.cgi
GET/POST /ptz/focus_type.cgi

parameters:

Name	Value	Description
act	get set	get current focus type. set the focus type

type	An integer	(only for act=set) the type of focus function
		type=1: auto focus
		type=0: manual focus

response:

Name	Value	Description

type	An integer	the type of focus function
		type=1: auto focus
		type=0: manual focus

5.1.14.adjust the focus manually, focus near or focus far from current position

request:

GET/POST /config/manual_focus.cgi
GET/POST /ptz/manual_focus.cgi

parameters:

Name	Value	Description
step	An integer (-32…32)	The step for focus near or focus far. The positive value means focus far (focus on distant object), the negative value means focus near (focus on near object).

response:

See the above table. If the movement is out of boundary, you will get the actual step performed. If current focus type is not manual focus, this CGI does nothing and returns "step=0"

5.1.15.get PTZ control privilege groups

This command allows you to query the list of privilege groups which could access /ptz/ directory to control PTZ and focus of camera.

request:
GET /config/ptz_privilege.cgi

response:

Name	Value	Description
groups	A string	List of groups separated by comma which are given the privilege to access /ptz/ directory to control PTZ and focus of camera. Administrator group are always accessible to /ptz/ directory, not matter is listed in this parameter or not.

5.1.16.set PTZ control privilege groups

This command allows you to configure the list of privilege groups which could access to /ptz/ directory to control PTZ and focus of camera.

request:
GET/POST /config/ptz_privilege.cgi

parameters:
See the above table.

response:
See the above table.

5.1.17.query focus information

request:
GET /config/focus_info.cgi

response:

Name	Value	Description
max	An integer	maximum position of focus.
min	An integer	minimum position of focus.

5.1.18.get the current position of focus

request:
GET /config/focus_pos.cgi

response:

Name	Value	Description
focus	An integer	The focus position.

5.1.19.set absolutely position of focus

request:
GET/POST /config/focus_pos.cgi
parameters:
see the above table.
response:
see the above table.

5.1.20.fine-tune focus automatically

request:
GET/POST /config/focus_act.cgi

parameters:

Name	Value	Description
autofocus	yes	Fine-tune focus automatically.

response:

Name	Value	Description
autofocus	yes, fail	The action was successful or fails.

5.1.21.PTZ direction of movement

request:
GET/POST /config/ptz_direction.cgi

parameters:

Name	Value	Description
direction	String	Use “up”、“down”、”left”、”right”、”upleft”、”upright”、”downleft”、”downright”、”stop”、”zoomwide”、”zoomtele”、”zoomstop”.
speed	An integer	speed control for PTZ (1 – 10)

5.2.via RS-485

5.2.1.do RS-485 commands

request:
GET/POST /config/rs485_do.cgi

parameters:

Name	Value	Description
direction	0-13	0: wide (zoom out) with stop. 1: up with stop 2: tele (zoom in) with stop 3: left with stop 4: home (only for custom command) 5: right with stop 6: focus far with stop 7: down with stop 8: focus near with stop 9: 10: custom command 1 11: custom command 2 12: custom command 3 13: custom command 4
speed	An integer	speed control for up, down, left, right. (1 – 10) (includes: Dyna, Lilin, Lilin2, PelcoD, PelcoP)

5.3.PTDC Pan/Tilt get information

5.3.1.get Pan/Tilt Position

Get the current Pan/Tilt position degree
request:
GET /cgi/ptdc.cgi?command=get_pos

Response: Represented by XML	<config> <posX>0</posX> <posY>0</posY> </config>

5.3.2.get Pan/Tilt Position by Step

Get the current Pan/Tilt position step
request:
    GET /cgi/ptdc.cgi?command=get_pos_step

Response: Represented by XML	<config> <posXStep>0</posXStep> <posYStep>0</posYStep> </config>

5.3.3.get Pan/Tilt Boundary

Get Pan/Tilt position degree boundary
request:
    GET /cgi/ptdc.cgi?command=get_boundary

Response: Represented by XML	<config> <boundaryMaxX>164</boundaryMaxX> <boundaryMinX>-164</boundaryMinX> <boundaryMaxY>83</boundaryMaxY> <boundaryMinY>-30</boundaryMinY> </config>

5.3.4.get Pan/Tilt Boundary by Step

Get Pan/Tilt position step boundary
request:
    GET /cgi/ptdc.cgi?command=get_step_boundary

Response: Represented by XML	<config> <boundaryMaxStepX>18944</boundaryMaxStepX> <boundaryMinStepX>-18944</boundaryMinStepX> <boundaryMaxStepY>14528</boundaryMaxStepY> <boundaryMinStepY>-5360</boundaryMinStepY> </config>

5.3.5.get Pan/Tilt Accuracy

Get the current Pan/Tilt minimum valid movement and precision degree.
request:
    GET /cgi/ptdc.cgi?command=get_pt_accuracy

Response: Represented by XML	<config> <panMinMovement>0.14</panMinMovement> <panPrecision>0.14</panPrecision> <tiltMinMovement>0.18</tiltMinMovement> <tiltPrecision>0.09</tiltPrecision> </config>

note:
xxxMinMovement: user has to set the minimum degree in every movement via Pan and/or Tilt to make sure the physical PT's movement visible.

xxxPrecision: the movement unit, user has to set the degree as multiple times of xxxPrecision for PT hardware precision consideration.

5.3.6.get Pan/Tilt Accuracy by Step

Get the current Pan/Tilt minimum valid movement and precision steps.
request:
GET /cgi/ptdc.cgi?command=get_pt_step_accuracy

Response: Represented by XML	<config> <panMinMovementStep>16</panMinMovementStep> <panPrecisionStep>16</panPrecisionStep> <tiltMinMovementStep>32</tiltMinMovementStep> <tiltPrecisionStep>16</tiltPrecisionStep> </config>

note:
xxxMinMovementStep: user has to set the minimum steps in every movement via Pan and/or Tilt to make sure the physical PT's movement visible.
xxxPrecisionStep: the movement unit, user has to set the steps as multiple times of PrecisionStep for PT hardware precision consideration.

5.3.7.get Pan/Tilt View Angle

Get Pan/Tilt view angle degree in live view.request:
GET /cgi/ptdc.cgi?command=get_view_angle

Response: Represented by XML	<config> <viewAngleHorizontal>51</viewAngleHorizontal> <viewAngleVertical>39</viewAngleVertical> </config>

5.3.8.get Pan/Tilt View Step

Get Pan/Tilt view angle degree in live view.request:
GET /cgi/ptdc.cgi?command=get_view_step

Response: Represented by XML	<config> <viewStepHorizontal>5865</viewStepHorizontal> <viewStepVertical>6747</viewStepVertical> </config>

5.3.9.get Pan/Tilt Preset Positions

Get the current Pan/tilt Preset Positions' information:
GET /cgi/ptdc.cgi?command=get_preset_positions

Response: Represented by XML	<config> <size>2</size> <presetName0>1</presetName0> <presetX0>0</presetX0> <presetY0>30</presetY0> <presetZ0>1</presetZ0> <presetF0>0</presetF0> <presetFType0>1</presetFType0> <presetName1>2</presetName1> <presetX1>-60</presetX1>

```
        <presetY1>30</presetY1>
        <presetZ1>1</presetZ1>
        <presetF1>0</presetF1>
        <presetFType1>1</presetFType1>
</config>
```

note:
X,Y   Unit: degree (pan/tilt)
Z   Unit:optical mag (zoom)
F   Unit:Step (focus)
FT   0: manual 1: auto (focus type)

5.3.10.get Pan/Tilt/Zoom Hardware information

Get the current Pan/Tilt/Zoom/Focus Hardware information:
GET /cgi/ptdc.cgi?command=get_ptz_hw_info

Response:
Represented by XML

```
<config>
    <pan>
            <inAngle>
                    <viewAngleHorizontal>45</viewAngleHorizontal>
                    <boundaryMaxX>164</boundaryMaxX>
                    <boundaryMinX>-164</boundaryMinX>
                    <panMinMovement>0.14</panMinMovement>
                    <panPrecision>0.14</panPrecision>
            </inAngle>
            <inStep>
                   <viewStepHorizontal>5175</viewStepHorizontal>
                    <boundaryMaxStepX>18944</boundaryMaxStepX>
                    <boundaryMinStepX>-18944</boundaryMinStepX>
                    <panMinMovementStep>16</panMinMovementStep>
                    <panPrecisionStep>16</panPrecisionStep>
            </inStep>
    </pan>
    <tilt>
        <inAngle>
                    <viewAngleVertical>38</viewAngleVertical>
                    <boundaryMaxY>83</boundaryMaxY>
                    <boundaryMinY>-30</boundaryMinY>
                    <tiltMinMovement>0.18</tiltMinMovement>
                    <tiltPrecision>0.09</tiltPrecision>
            </inAngle>
            <inStep>
                   <viewStepVertical>6574</viewStepVertical>
                   <boundaryMaxStepY>14528</boundaryMaxStepY>
                   <boundaryMinStepY>-5360</boundaryMinStepY>
                   <tiltMinMovementStep>32</tiltMinMovementStep>
                   <tiltPrecisionStep>16</tiltPrecisionStep>
            </inStep>

    </tilt>
    <zoom>
        <inMag>
                    <boundaryMaxZoom>10</boundaryMaxZoom>
                    <boundaryMinZoom>1</boundaryMinZoom>
                    <zoomMinMovement>0.09</ zoomMinMovement>
                    <zoomPrecision>0.09</ zoomPrecision>
        </inMag>
        <inStep>
                    <zoomMaxStep>1174</ zoomMaxStep>
                    <zoomMinStep>0</zoomMinStep>
                    <zoomMinMovementStep>1</zoomMinMovementStep>
                    <zoomPrecisionStep>1</zoomPrecisionStep>
        </inStep>
    </zoom>
    <focus>
            <boundaryMaxFocus>1038</boundaryMaxFocus>
```

	<boundaryMinFocus>194</boundaryMinFocus> <focusMinMovementStep>1</focusMinMovementStep> <focusPrecisionStep>1</focusPrecisionStep> </focus> </config>

note:
Pan/Tilt Unit: degree
Zoom :
inMag Unit: optical mag
Zoom :
inStep Unit: step
Focus Unit: step

5.3.11.get Pan/Tilt Home Position

Get the current home Pan/Tilt position:
GET /cgi/ptdc.cgi?command=get_home_pos

Response: Represented by XML	<config> <homePosX>0</homePosX> <homePosY>0</homePosY> </config>

5.3.12.get Pan/Tilt Patrol Speed

Get the patrol speed in patrol action:
GET /cgi/ptdc.cgi?command=get_patrol_speed

Response: Represented by XML	<config> <speed>5</speed> </config>

5.3.13.get Pan/Tilt Wait Time

Get the wait_time in patrol action:
GET /cgi/ptdc.cgi?command=get_wait_time

Response: Represented by XML	<config> <waitTime>5</waitTime> </config>

note:
Unit: second
when patrol to a turning point, it will stop and wait for “wait_time” seconds

5.4.PTDC Pan/Tilt set information

5.4.1.set Pan/Tilt Position

Set the Pan/Tilt absolute position in degree unit:

request:

GET/POST /cgi/ptdc.cgi?command=set_pos&posX=100&posY=30

parameters:

Name	Value	Description
posX	float	The movement degree is based on Zoom mag 1x
posY	float	The movement degree is based on Zoom mag 1x

response:

A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	Need correct posX, posY parameters
	3. failure	Error command type
	4. oob	Out Of Boundary

5.4.2.set Pan/Tilt Relative Position

Set the Pan/Tilt relative position in degree unit:
request:

GET/POST /cgi/ptdc.cgi?command=set_relative_pos&posX=100&posY=30

parameters:

Name	Value	Description
posX	float	The movement degree is based on Zoom mag 1x
posY	float	The movement degree is based on Zoom mag 1x

response:

A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	Need correct posX, posY parameters
	3. failure	Error command type
	4. oob	Out Of Boundary

5.4.3.set Pan/Tilt Position by Step

Set the Pan/Tilt absolute position in step unit:
request:

GET/POST /cgi/ptdc.cgi?command=set_pos_step&posXStep=100&posYStep=30

parameters:

Name	Value	Description
posXStep	#	The movement step is not need to base on Zoom mag 1x
posYStep	#	The movement step is not need to base on Zoom mag 1x

response:

A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	Need correct posXStep, posYStep parameters
	3. failure	Error command type
	4. oob	Out Of Boundary

5.4.4.set Pan/Tilt Relative Position by Step

Set the Pan/Tilt relative position in step unit:
request:

```
GET/POST /cgi/ptdc.cgi?command=set_relative_pos_step&posXStep=100&posYStep=30
```

parameters:

Name	Value	Description
posXStep	#	The movement step is not need to base on Zoom mag 1x
posYStep	#	The movement step is not need to base on Zoom mag 1x

response:
A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	Need correct posXStep, posYStep parameters
	3. failure	Error command type
	4. oob	Out Of Boundary

5.4.5.set Home

Set the current Pan/Tilt/Zoom/Focus/FocusType as home position:
request:

```
GET/POST /cgi/ptdc.cgi?command=set_home
```

response:
A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.4.6.Restore Default Home

Restore to the factory default home position:
request:

```
GET/POST /cgi/ptdc.cgi?command=restore_home
```

response:
A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.4.7.set Patrol Speed

Set the patrol speed(single_pan/pan_patrol/single_patrol/user_patrol):
request:

GET/POST /cgi/ptdc.cgi?command=set_patrol_speed&speed=3

parameters:

Name	Value	Description
speed	#	Patrol Speed(1,2,3…)

response:

A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	Need correct speed parameter
	3. failure	Error command type
	4. oob	Out Of Boundary

5.4.8.set Patrol Wait Time

Set the wait time for patrol action.
request:

GET/POST /cgi/ptdc.cgi?command=set_wait_time&waitTime=3

parameters:

Name	Value	Description
waitTime	#	between 0 ~ 3600 sec

response:

A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	Need correct waitTime parameter
	3. failure	Error command type
	4. oob	Out Of Boundary

5.5.PTDC Pan/Tilt other Parts

5.5.1.Calibration

Pan/Tilt calibration, and then move to current home position.
request:

```
GET/POST /cgi/ptdc.cgi?command=calibration
```

response:
A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.2.Single Pan

Pan/Tilt patrol along pan direction once.
request:

```
GET/POST /cgi/ptdc.cgi?command=single_pan
```

response:
A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.3.Pan Patrol

Pan/Tilt patrol along pan direction for user define times.
request:

```
GET/POST /cgi/ptdc.cgi?command=pan_patrol
```

response:
A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.4.Single Patrol

Pan/Tilt patrol depend on user define route once.
request:

```
GET/POST /cgi/ptdc.cgi?command=single_patrol
```

response:
A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.5.User Patrol

Pan/Tilt patrol depend on user define route and times.
request:

```
GET/POST /cgi/ptdc.cgi?command=user_patrol
```

response:
A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.6.Stop Patrol

soft stop(it will not stop immediately until reach a turning point)
request:

```
GET/POST /cgi/ptdc.cgi?command=stop_patrol
```

response:
A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.7.Stop

Hard stop (Pan/Tilt will stop immediately)
request:

```
GET/POST /cgi/ptdc.cgi?command=stop
```

response:
A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success

5.5.8.Go Home

Move the Pan/Tilt to home position:
request:

```
GET/POST /cgi/ptdc.cgi?command=go_home
```

response:
A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>

Result code	1. ok	Success

5.5.9.Goto Preset Position

Go to preset point as user define (index 0 is the first point as user define instead of index1) (zero base)
request:

```
GET/POST /cgi/ptdc.cgi?command=goto_preset_position&index=0

or

GET/POST /cgi/ptdc.cgi?command=goto_preset_position&presetName=aaa
```

response:
A result tag will also be presented:

Result:
Represented by XML
	`<result>`
	`<code>ok</code>`
	`</result>`	

Result code	1. ok	Success
	2. invalidParameter	Need correct index or presetName parameter
	3. failure	Error command type

5.6.PTDC Zoom/Focus/Focus Type get information

5.6.1.get Zoom Boundary

Get Zoom mag boundary, 1 represent as 1x mag, 10 represent as 10x mag.
request:

```
GET /cgi/ptdc.cgi?command=get_zoom_boundary
```

Response:
Represented by XML
	`<config>`
	`<boundaryMaxZoom>10</boundaryMaxZoom>`
	`<boundaryMinZoom>1</boundaryMinZoom>`
	`</config>`

note:
Unit: mag

5.6.2.get Zoom Mag

Get the current Zoom mag.
request:

```
GET /cgi/ptdc.cgi?command=get_zoom_mag
```

Response:
Represented by XML
	`<config>`
	`<zoomMag>5.5</zoomMag>`
	`</config>`

note:
The upper return value represent the current optical mag is 5.5x

5.6.3.get Zoom Boundary by Step

Get Zoom step boundary.
request:

```
GET /cgi/ptdc.cgi?command=get_zoom_step_boundary
```

Response:
Represented by XML
	<config>
	        <zoomMaxStep>1174</ zoomMaxStep>
	        <zoomMinStep 0</ zoomMinStep>
	</config>

5.6.4.get Zoom Step

Get the current Zoom step.
request:
    GET /cgi/ptdc.cgi?command=get_zoom_step

Response: Represented by XML	<config> <zoomStep>1174</zoomStep> </config>

note:
Unit: step
The upper return value represent the current optical step is 1174

5.6.5.get Focus Boundary

Get Focus boundary.
request:
    GET /cgi/ptdc.cgi?command=get_focus_boundary

Response:
Represented by XML
	<config>
	        <boundaryMaxFocus>1024</boundaryMaxFocus>
	        <boundaryMinFocus>194</boundaryMinFocus>
	</config>

note:
Unit: step
The upper return value represent the focus step range can be set.

5.6.6.get Focus Step

Get the current Focus step.
request:
    GET /cgi/ptdc.cgi?command=get_focus_step

Response: Represented by XML	<config> <focusStep>1000</focusStep> </config>

5.6.7.get Focus Type

Get the current Focus Type.
request:
    GET /cgi/ptdc.cgi?command=get_zoom_boundary

Response: Represented by XML	<config> <focusType>1</focusType> </config>

note:

focusType 0:manual focus 1:auto focus

5.6.8.get Zoom Accuracy

Get the current Zoom minimum valid movement and precision mag.
request:
GET /cgi/ptdc.cgi?command=get_zoom_accuracy

Response: Represented by XML	<config> <zoomMinMovement>0.09</zoomMinMovement> <zoomPrecision>0.09</zoomPrecision> </config>

note:
Unit: mag
xxxMinMovement: user has to set the minimum mag in every movement via Zoom to make sure the physical Zoom's movement visible.
xxxPrecision: the movement unit, user has to set the mag as multiple times of xxxPrecision for Zoom hardware precision consideration.

5.6.9.get Focus Type

Get the current Zoom minimum valid movement and precision steps.
request:
GET /cgi/ptdc.cgi?command=get_zoom_step_accuracy

Response: Represented by XML	<config> <zoomMinMovementStep>1</zoomMinMovementStep> <zoomPrecisionStep>1</zoomPrecisionStep> </config>

note:
Unit: step
xxxMinMovementStep: user has to set the minimum steps in every movement via Zoom to make sure the physical Zoom's movement visible.
xxxPrecisionStep: the movement unit, user has to set the steps as multiple times of PrecisionStep for Zoom hardware precision consideration.

5.6.10.get Focus Accuracy

Get the current focus minimum valid movement and precision steps.
request:
GET /cgi/ptdc.cgi?command=get_focus_accuracy

Response: Represented by XML	<config> <focusMinMovementStep>1</focusMinMovementStep> <focusPrecisionStep>1</focusPrecisionStep> </config>

note:
Unit: step
xxxMinMovementStep: user has to set the minimum steps in every movement via Focus to make sure the physical Focus's movement visible.
xxxPrecisionStep: the movement unit, user has to set the steps as multiple times of PrecisionStep for Focus hardware precision consideration.

5.6.11.get Home Zoom Mag

Get the current home Zoom mag.

request:

```
GET /cgi/ptdc.cgi?command=get_home_zoom_mag
```

Response: Represented by XML	<config> <homeZoomMag>5.5</homeZoomMag> </config>

note:
Unit: mag
The upper return value represent the home optical mag is 5.5x

5.6.12.get Home Focus Step

Get the current home Zoom step.
request:

```
GET /cgi/ptdc.cgi?command=get_home_zoom_step
```

Response: Represented by XML	<config> <homeZoomStep>1174</homeZoomStep> </config>

note:
Unit: step
The following return value represent the home optical step is 1174

5.6.13.get Home Focus Step

Get the current home focus step.
request:

```
GET /cgi/ptdc.cgi?command=get_home_focus_step
```

Response: Represented by XML	<config> <homeFocusStep>1000</homeFocusStep> </config>

note:
Unit: step
The following return value represent the home focus step is 1000

5.6.14.get Home Focus Type

Get the current home focus type
request:

```
GET /cgi/ptdc.cgi?command=get_home_focus_type
```

Response: Represented by XML	<config> <homeFocusType>1</homeFocusType> </config>

note:
focusType 0:manual focus 1:auto focus

5.7.PTDC Zoom/Focus/Focus Type set information

5.7.1.set Zoom Mag

Set the Zoom absolute position in mag unit:
request:

```
GET/POST /cgi/ptdc.cgi?command=set_zoom&zoom_mag=5.5
```

parameters:

Name	Value	Description
zoom_mag	float	Zoom mag

response:
A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	Need correct zoom_mag parameters
	3. failure	Error command type
	4. oob	Out Of Boundary

5.7.2.set Relative Zoom Mag

Set the relative Zoom position in mag unit:
request:

```
GET/POST /cgi/ptdc.cgi?command=set_relative_zoom&zoom_mag=2.5
```

parameters:

Name	Value	Description
zoom_mag	float	Zoom mag

response:
A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	Need correct zoom_mag parameters
	3. failure	Error command type
	4. oob	Out Of Boundary

Note:

Unit: mag
Add more 2.5x mag to current Zoom mag, ex original mag is 5.5x, and then goes to 8.0x after this command.

5.7.3.set Zoom Step

Set the absolute Zoom position in step unit:
request:

```
GET/POST /cgi/ptdc.cgi?command=set_zoom_step&zoom_step=50
```

parameters:

Name	Value	Description
zoom_step	#	Zoom step

response:

A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	Need correct zoom_step parameters
	3. failure	Error command type
	4. oob	Out Of Boundary

5.7.4.set Relative Zoom Step

Set the relative Zoom position in step unit:

request:

```
GET/POST /cgi/ptdc.cgi?command=set_relative_zoom_step&zoom_step=25
```

parameters:

Name	Value	Description
zoom_step	#	Zoom step

response:

A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	Need correct zoom_step parameters
	3. failure	Error command type
	4. oob	Out Of Boundary

Note:

Unit: step

Add more 25 step to current Zoom step, ex original Zoom step is 50, and then goes to 75 after this command.

5.7.5.set Focus Step

Set the Focus absolute position in step unit:

request:

```
GET/POST /cgi/ptdc.cgi?command=set_focus&focus_step=500
```

parameters:

Name	Value	Description
focus_step	#	Focus step

response:

A result tag will also be presented:

Result: Represented by XML	<result> <code>ok</code> </result>	
Result code	1. ok	Success
	2. invalidParameter	Need correct focus_step parameters
	3. failure	Error command type
	4. oob	Out Of Boundary

5.7.6.set Relative Focus Step

Set the Focus relative position in step unit:
request:

```
GET/POST /cgi/ptdc.cgi?command=set_relative_focus_step&focus_step=25
```

parameters:

Name	Value	Description
focus_step	#	Focus step

response:

A result tag will also be presented:

Result:
Represented by XML
	`<result>`
	`<code>ok</code>`
	`</result>`	

Result code	1. ok	Success
	2. invalidParameter	Need correct focus_step parameters
	3. failure	Error command type
	4. oob	Out Of Boundary

Note:

Unit: step
Add more 50 step to current Focus step, ex original Focus step is 250, and then goes to 300 after this command.

5.7.7.set Focus Type

Set Focus type as manual or auto mode:
request:

```
GET/POST /cgi/ptdc.cgi?command=set_focus_type&focus_type=0
```

parameters:

Name	Value	Description
focus_type	0 or 1	Focus Type

response:

A result tag will also be presented:

Result:
Represented by XML
	`<result>`
	`<code>ok</code>`
	`</result>`	

Result code	1. ok	Success
	2. invalidParameter	Need correct focus_type parameter
	3. failure	Error command type
	4. oob	Out Of Boundary

Note:

Focus_type 0:manual focus 1:auto focus

5.8.Digital PTZ Control

5.8.1.get the current digital PTZ position

request:

```
GET /config/digital_ptz_pos.cgi
```

response:

Name	Value	Description

p	An integer	The digital pan position.
t	An integer	The digital tilt position.
z	An integer	The digital zoom position.

5.8.2.add, delete or goto a digital PTZ preset

request:
GET /config/digital_ptz_preset.cgi

parameters:

Name	Value	Description
profileid	#	Profile number(# is a number from 1 to count of profiles)
act	add, del, go	Action type.
name	A String	Digital preset name

response:
see the above table. If the movement is out of boundary, you will get the actual absolute position.

5.8.3.move digital PTZ absolutely

request:
GET /config/digital_ptz_pos.cgi

parameters:

Name	Value	Description
p	An integer	Pans the device relative to the (0,0,0) position
t	An integer	Tilts the device relative to the (0,0,0) position
z	An integer	Zooms the device relative to the (0,0,0) position.

response:
see the above table. If the movement is out of boundary, you will get the actual absolute position.

5.8.4.move digital PTZ relatively

request:
GET /config/digital_ptz_move_rel.cgi

parameters:

Name	Value	Description
profileid	#	Profile number(# is a number from 1 to count of profiles)
p	An integer	Pans the device relative to the current position
t	An integer	Tilts the device relative to the current position
z	An integer	Zooms the device relative to the current position.

response:
see the above table. If the movement is out of boundary, you will get the actual relative p, t, z values it move.

5.8.5.digital PTZ autopan

request:
GET /config/digital_ptz_autopan.cgi

parameters:

Name	Value	Description
profileid	#	Profile number(# is a number from 1 to count of profiles)
act	go, stop	Action type.

response:

see the above table. If the movement is out of boundary, you will get the actual relative p, t, z values it move.

5.8.6.digital PTZ sequence

request:

GET /config/digital_ptz_sequence.cgi

parameters:

Name	Value	Description
profileid	#	Profile number(# is a number from 1 to count of profiles)
act	add, del, go, stop	Action type.
name	A string	Only for “act=add”
index	An integer	From 0 to max
time	An integer	Only for “act=add”

response:

see the above table. If the movement is out of boundary, you will get the actual relative p, t, z values it move.

6.Notification API

6.1.Camera status notification

6.1.1.get the notification status

This CGI is an one-shot command, which only return current status of IP camera. If you need to minitor camera status for a long time, please use 6.1.2. notify_stream.cgi instead.

request:

```
GET /config/notify.cgi
```

response:

Name	Value	Description
md#	on, off	event motion detection # is triggered or not.
mdv#	<degree of motion> (0-100)	Percentage of motion detected by camera.
input#	on, off	event input # is triggered or not.
storagefull	on, off	event storage full
storagefail	on, off	event storage fail
recording	on, off	status is recording
snapshooting	on, off	status is snapshooting
output#	on, off	status of output # is on or off
vsignal	on, off	status of video signal is on or lost
speaker	on, off	status of speaker is on or off
mic	on, off	status of microphone is on or off
mdetecting	on, off	setting of motion detection is on or off
irled	on, off	status of IR LED is on or off
autofocusbusy	yes, no	Status of autofocus is on or off

6.1.2.get the notification stream

request:

```
GET /config/notify_stream.cgi
```

response:

The client side should keep receiving notification information from camera. It includes all available events or status as follow table. The notification information is only generated on while event or status changed. If there is no any changed event or status being reported within 30 second, a special tag: “`keep_alive`” will be send to the client side.

Name	Value	Description
md#	on, off	event motion detection # is triggered or not.
mdv#	<degree of motion> (0-100)	Percentage of motion detected by camera.
input#	on, off	event input # is triggered or not.
storagefull	on, off	event storage full
storagefail	on, off	event storage fail
recording	on, off	status is recording
snapshooting	on, off	status is snapshooting
output#	on, off	status of output # is on or off
vsignal	on, off	status of video signal is on or lost
speaker	on, off	status of speaker is on or off
mic	on, off	status of microphone is on or off
usbstatus	ready, protected, full, invalid	status of SD card inserted in the camera.

cameraname	<camera name>	status of camera name
irled	on, off	status of IR LED is on or off
autofocusbusy	yes, no	Status of autofocus is on or off

7.RTSP API

The Real-Time Streaming Protocol (RTSP) is a protocol to get audio and video streaming data provided by a media server. IP camera can act as a media server and stream the real time audio and video . By RTSP request, a client application can get streaming data from IP camera. The detail of RTSP protocol please refer to RFC 2326.

7.1.Live streaming

7.1.1.get URL entry of specified profile

request:

GET /config/rtspurl.cgi

parameters:

profileid=<video profile number>

response:

Name	Value	Description
profileid	#	profile number (# is a number from 1 to the count of profiles)
urlentry	<entry of video profile>	URL entry of associated video stream profile

7.1.2.set video config

request:

GET/POST /config/rtspurl.cgi

parameters:

see the above table.

response:

see the above table.

7.1.3.Get live video

The requested URI of an IP camera stream data can be described by following:

rtsp://<server ip>/<urlentry>

Get video and audio stream for use on PC.

Where <urlentry> is the url entry associated with one of the video profile. The value can be got by calling **/config/rtspurl.cgi** (see 7.1.1)

NOTE:

Since our camera now can let user to change the url entry of each video profile, the following RTSP urls are obsolete, user should use 7.1.3 to get RTSP stream.

rtsp://<server ip>/mp4

Get video and audio stream with MPEG-4 video format for use on PC.

rtsp://<server ip>/jpeg

Get video (and audio) stream with M-JPEG video format for use on PC.

rtsp://<server ip>/3gpp

Get video (and audio) stream with MPEG-4 video format for use on 3GPP compliant device.

rtsp://<server ip>/live#

where # is the number from 1 to the count of video profile. For example, use rtsp://192.168.1.1/live1 to get the stream of video profile number 1.

7.2 RTSP Methods:

A. OPTIONS: Report the methods supported by the IP camera.

Please use "OPTIONS" method to get the other methods supported by the IP camera.

8.Appendix

8.1.Table used in NIPCA

Table 1: Time zone

ID	Time zone
1	(GMT-12:00) International Date Line West
2	(GMT-11:00) Samoa
3	(GMT-10:00) Hawaii
4	(GMT-09:00) Alaska
5	(GMT-08:00) Pacific Time (US & Canada)
6	(GMT-08:00) Tijuana, Baja California
7	(GMT-07:00) Chihuahua, La Paz, Mazatlan
8	(GMT-07:00) Mountain Time (US & Canada)
9	(GMT-07:00) Arizona
10	(GMT-06:00) Central America
11	(GMT-06:00) Guadalajara, Mexico City, Monterrey
12	(GMT-06:00) Saskatchewan
13	(GMT-06:00) Central Time (US & Canada)
14	(GMT-05:00) Bogota, Lima, Quito
15	(GMT-05:00) Eastern Time (US & Canada)
16	(GMT-05:00) Indiana (East)
17	(GMT-04:00) La Paz, Georgetown
18	(GMT-04:00) Atlantic Time (Canada)
19	(GMT-04:00) Santiago
20	(GMT-04:00) Manaus
21	(GMT-03:30) Newfoundland
22	(GMT-03:00) Buenos Aires
23	(GMT-03:00) Brasilia
24	(GMT-03:00) Greenland
25	(GMT-03:00) Montevideo
26	(GMT-02:00) Mid-Atlantic
27	(GMT-01:00) Azores
28	(GMT-01:00) Cape Verde Is.
29	(GMT) Greenwich Mean Time : Dublin, Edinburgh, Lisbon, London
30	(GMT) Monrovia, Reykjavik
31	(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana, Prague
32	(GMT+01:00) West Central Africa
33	(GMT+01:00) Sarajevo, Skopje, Warsaw, Zagreb
34	(GMT+01:00) Brussels, Copenhagen, Madrid, Paris
35	(GMT+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
36	(GMT+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius
37	(GMT+02:00) Athens, Bucharest, Istanbul
38	(GMT+02:00) Beirut
39	(GMT+02:00) Harare, Pretoria
40	(GMT+02:00) Cairo
41	(GMT+03:00) Minsk
42	(GMT+02:00) Amman
43	(GMT+01:00) Windhoek
44	(GMT+02:00) Jerusalem
45	(GMT+03:00) Baghdad

ID	Time zone
46	(GMT+03:00) Moscow, St. Petersburg, Volgograd
47	(GMT+04:00) Tbilisi
48	(GMT+03:00) Nairobi
49	(GMT+03:00) Kuwait, Riyadh
50	(GMT+03:30) Tehran
51	(GMT+04:00) Baku
52	(GMT+04:00) Abu Dhabi, Muscat
53	(GMT+04:00) Yerevan
54	(GMT+04:30) Kabul
55	(GMT+06:00) Yekaterinburg
56	(GMT+05:00) Islamabad, Karachi, Tashkent
57	(GMT+05:30) Chennai, Kolkata, Mumbai, New Delhi
58	(GMT+05:30) Sri Jayawardenepura
59	(GMT+05:45) Kathmandu
60	(GMT+06:00) Astana, Dhaka
61	(GMT+07:00) Novosibirsk
62	(GMT+06:30) Yangon (Rangoon)
63	(GMT+08:00) Krasnoyarsk
64	(GMT+07:00) Bangkok, Hanoi, Jakarta
65	(GMT+08:00) Beijing, Chongqing, Hong Kong, Urumqi
66	(GMT+08:00) Taipei
67	(GMT+09:00) Irkutsk, Ulaan Bataar
68	(GMT+08:00) Perth
69	(GMT+08:00) Kuala Lumpur, Singapore
70	(GMT+10:00) Yakutsk
71	(GMT+09:00) Osaka, Sapporo, Tokyo
72	(GMT+09:00) Seoul
73	(GMT+09:30) Adelaide
74	(GMT+09:30) Darwin
75	(GMT+10:00) Hobart
76	(GMT+10:00) Brisbane
77	(GMT+10:00) Vladivostok
78	(GMT+10:00) Canberra, Melbourne, Sydney
79	(GMT+10:00) Guam, Port Moresby
80	(GMT+12:00) Magadan
81	(GMT+12:00) Fiji
82	(GMT+12:00) Auckland, Wellington
83	(GMT+13:00) Nukualofa
84	(GMT-04:30)Caracas
85	(GMT+11:00) Solomon Is., New Caledonia
86	(GMT) Casablanca
87	(GMT+08:00) Ulaanbaatar

Table 2: dynamic DNS service providers

ID	provider URIs
	www.ez-ip.net
	www.penguinpowered.com
	members.dhs.org
dyndns	members.dyndns.org
	www.3322.org
	update.ods.org
	cgi.tzo.com
	members.easydns.com
	api.easydns.com

	www.justlinux.com
	www.dyns.cx
	dup.hn.org
	www.zoneedit.com
	ipv6tb.he.net

8.2.Advanced IP-Camera Stream (ACS) Header

Multimedia header:

ACS Audio header	ACS Video header

typedef struct _ACS_AudioHeader
{
unsigned long ulHdrID; //Header ID
unsigned long ulHdrLength;
unsigned long ulDataLength;
unsigned long ulSequenceNumber;
unsigned long ulTimeSec;
unsigned long ulTimeUSec;
unsigned long ulDataCheckSum;
unsigned short usFormat;
unsigned short usChannels;
unsigned short usSampleRate;
unsigned short usSampleBits;
unsigned long ulReserved;
}ACS_AudioHeader, *PACS_AudioHeader;
	typedef struct _ACS_VideoHeader
	{
	unsigned long ulHdrID; //Header ID
	unsigned long ulHdrLength;
	unsigned long ulDataLength;
	unsigned long ulSequenceNumber;
	unsigned long ulTimeSec;
	unsigned long ulTimeUSec;
	unsigned long ulDataCheckSum;
	unsigned short usCodingType;
	unsigned short usFrameRate;
	unsigned short usWidth;
	unsigned short usHeight;
	unsigned char ucMDBitmap;
	unsigned char ucMDPowers[3];
	}ACS_VideoHeader, *PACS_VideoHeader

Description:

The byte order of this header is little-endian.

Common part:

`ulHdrID`: Special id for identifying ACS header. For audio: the value of this id is 0xF6010000 (since our header is in little-endian so the byte array of this id is ’00 00 01 F6’). For video the value is 0xF5010000.

`ulHdrLength`: Header length. (32 bytes in current version)

`ulDataLength`: Payload data length.

`ulSequenceNumber`: Sequence number.

`ulTimeSec`: Time stamp in sec since 1970/01/01 00:00.

`ulTimeUSec`: Microsecond part of time stamp

`ulDataCheckSum`: Store last 4 bytes of payload data.

Audio part:

`usFormat`: Audio data format. The possible value:

```
AFMT_MS_ADPCM: 0

AFMT_MU_LAW: 1
AFMT_A_LAW: 2
AFMT_IMA_ADPCM: 4
AFMT_U8: 8
AFMT_S16_LE: 0x10 /* Little endian signed 16*/
AFMT_S16_BE: 0x20 /* Big endian signed 16 */
AFMT_S8: 0x40
AFMT_U16_LE: 0x80 /* Little endian U16 */
AFMT_U16_BE: 0x100 /* Big endian U16 */
AFMT_MPEG: 0x200 /* MPEG (2) audio */
AFMT_AC3: 0x400

AFMT_AMR: 0x800
```

AFMT_AAC: 0x1000

AFMT_ALAW: 0x2000

usChannels: Audio channels number: mono(1) or stereo(2).

usSampleRate: Sample rate.

usSampleBits: Bits count per sample.

ulReseverd: Reserved.

Video part:

usCodingType: Encoding type of frame. The possible values are:

VFCT_IVOP (MPEG4): 0

VFCT_PVOP (MPEG4): 1

VFCT_JPEG: 5

VFCT_H264_IFRM: 10

VFCT_H264_PFRM: 11

usFrameRate: Frames per second.

usWidth: The width of frame dimension

usHeight: The height of frame dimension

ucMDBitmap: The height of frame dimension

ucMDPowsers[3]: The height of frame dimension

Extension header:

We propose add extensive header for dealing with other information attaching with the multimedia stream. Instead of appending this kind of information to multimedia stream, it can save more bandwidth utilization.

Table: Alphanetworks IP-Camera streaming (ACS) extension header:

ACS extension header

typedef struct _ACS_ExtentHeader
{
unsigned long ulHdrID; // ’00 00 01 FE’
unsigned long ulHdrLength;
unsigned char pbyReserved[96];
} ACS_ExtentHeader, *PACS_ExtentHeader

Description:

The extension header is interleaved within the video stream or audio stream when the information is required by client.

ulHdrID: Special id for identifying ACS header. 0xFE010000.

ulHdrLength: Header length. (32 bytes in current version)

pbyReserved[96]: To be defined.