Gadspot NC1000 IP Camera Programmers API Reference.

Brett Smith

Mike Ubell

Gadspot NC1000 IP Camera Programmers API Reference.

Brett Smith
Mike Ubell

Table of Contents

Chapter 1. Introduction

1. About This Document

These notes are an attempt to document the API of the Gadspot NC1000 IP Camera. From the site (http://gadspot.com) ...

> Gadspot IP camera NC1000 features a standalone system, which means a camera and a computer combined in a small package. Built-in capture module and web server provide users to view images through IP network(e.g. LAN and WAN) with a browser on any computer connected to network.

The camera used was the wireless model (NC1000-W10) although most of this probably applies to the wired variant too.

Although the camera has plenty of features, the web interface for users of alternative operating systems and browsers is pretty poor. IE users get a more passable experience with the use of ActiveX controls instead of the applet / poorly designed HTML the rest of us see.

This document was born out the idea that this quite neat little device could be made easier to use.

It assumes you are familiar with the HTTP [http://www.w3.org/Protocols/] protocol and have some method of communicating via HTTP with the device. Many programming languages have built in support for the HTTP protocol, and failing that will have some external library available.

The firmware version(?) (as reported by GetVer.cgi - see below) is "Jul 8 2004 09:53:50".

2. General Notes

- Any detail marked with a (?) indicates the full details are unknwon. Anyone with any new information, pointers or corrections please feel free to contact me and Ill be happy to update this document.
- The camera has two user interfaces. One ActiveX based which will only work using Internet Explorer on Windows. For all other browsers there is a Java Applet version.
- If you mess up the camera settings, looking directly at the lens, there is a tiny reset in the bottom left hand side of the cover. Holding the button down for 10 seconds and restarting seemed to do the trick and restored all factory defaults..
- The `Server` HTTP header returned by the camera is 'WYM/1.0.

Chapter 2. API

1. Grabbing a Single Frame

Retrieving a single frame from the camera is the simplest and may well be all you wish to do.

1.1. Request

There are two methods of requesting a single frame.

1. *Current Frame*. Send a single HTTP request to

   ```
   http://<host>[:port]/Jpeg/CamImg.jpg
   ```

2. *Indexed Frame*. The camera appears to buffer a number of images (I have no idea how many) which may be retrieved using an index number. Send a single HTTP request to

   ```
   http://<host>[:<port>]/Jpeg/CamImg<ms since 1970><random number><frame index>.jpg
   ```

   The current frame index may be retrieved using the GetStatus command (see below).

1.2. Response

The camera will respond with the JPEG data. The `Content-Type` of the response will be `image/jpeg`.

2. CGI Commands

Now for the more interesting stuff. The camera supports many commands allowing you to get information about the device, configure it and alter the properties of the images produced.

Commands are sent to the camera as an HTTP request in the format :-

```
http://<host>[:<port>]/<command>.cgi
```

Some commands will also accept parameters as part of the request. When the request is sent using the GET method, the parameters are added to the request URI.

```
http://<host>[:port]/<command>.cgi?<name1>=<value1>&<name2>=<value2>.....
```

Depending on the command, the response to various requests will return different content of different content types.

Some responses will return lists of name / value pairs. These are returned as a response body of type text/plain in the format :-

```
<name1><space>=<space><value1>[CR]
<name2><space>=<space><value2>[CR]
..
..
<nameX><space>=<space><value2>[CR]
```

Other commands ('Set' commands for instance) will just return an empty body regardless of whether the command completed successfully.

2.1. GetData

Starts the camera streaming JPEG images.

AFAICS, the cameras only stream available is a simple sequence of JPEG images. This is what the specs say is MJPEG? I can find no concrete information on what MJPEG is other than its not really a true standard.

2.1.1. Parameters

None known.

2.1.2. Response

First off, a Content-Type HTTP header of 'multipart/x-mixed-replace;boundary=IPCamBoundary' is returned. The response body consists of many (infinite) parts delimited by this boundary string :-

```
--IPCamBoundary[CR]
Content-Type: image/jpeg[CR]
[CR]
<Jpeg image data 1>
--IPCamBoundary[CR]
Content-Type: image/jpeg[CR]
[CR]
<Jpeg image data 2>
..
..
..
--IPCamBoundary[CR]
Content-Type: image/jpeg[CR]
[CR]
<Jpeg image data X>
```

This repeats until the client closes the connection. Note, there is no [CR] between <Jpeg iamge data X> and the next --IPCamBoundary.

2.2. GetStatus

Gets a number of camera settings.

2.2.1. Parameters

None known.

2.2.2. Response

Response returns as the HTTP body (content type of text/plain) and is a single line consisting of fixed width fields. Integers are padded with zeroes.

```
<field1><field2><field3>.... etc etc ...<field13>
```

FIeld	Position	Type	Description
1	0-1	Integer	Camera Status (0=Closed,1=Open,<X>=?)
2	2-3	Integer	Modem Status (0=Disconnected,1=Connected,<X>=?).
3	4-5	Integer	PPPOE Status (0=Disconnected,1=Connected,<X>=?).
4	6-20	?????	?????
5	21	Integer	Resolution (0=176x144,1=352x288,2=320x240,3=640x480)

FIeld	Position	Type	Description
6	22	Integer	Quality (0=Lowest,4=Highest)
7	23	Integer	Privilege. (0=Admin,1=User,<X>=Unknown)
8	24-29	Integer	Frame Index
9	30-111	?????	?????

2.3. GetTime

Gets the current time and other time-related settings.

2.3.1. Parameters

None known.

2.3.2. Response

Six values are returned as name / value pairs (see 'General notes') in the following order :-

Name	Type	Description
Sec	Integer	Seconds since 1970.
TimeZone	Integer	Time zone.
NtpServer	String	Address of NTP server.
UseNtp	Boolean	0 = No, 1 = Yes.
DayLight	Boolean	0 = No, 1 = Yes.
Interval	Integer	Frequency in days to check time.

2.4. GetName

Gets the camera name.

2.4.1. Parameters

None known.

2.4.2. Response

The response consists of a single name / value pair:-

Name	Type	Description
Sec	Integer	Seconds since 1970.
TimeZone	Integer	Time zone.
NtpServer	String	Address of NTP server.
UseNtp	Boolean	0 = No, 1 = Yes.
DayLight	Boolean	0 = No, 1 = Yes.
Interval	Integer	Frequency in days to check time.

2.5. GetMotionDetect

Gets the state for motion detection.

2.5.1. Parameters

None known.

2.5.2. Response

The response consists of 2 name / value paris in the following order.

Name	Type	Description
Enable	Integer	Motion detection enabled (0=Yes, 1=No).
Sensitivity	Integer	0,1 or 2

2.6. GetMail

Gets the email settings for motion detection notification.

2.6.1. Parameters

None known.

2.6.2. Response

The response consists of 10 name / value paris in the following order.

Name	Type	Description
MailServer	String	IP (/ hostname?) of the email server.
Sender	String	Email address of the sender.
Receiver	String	Email address of the receiver.
Receiver_Cc	String	Email address for a carbon copy.
Receiver_Bcc	String	Email address for a blind carbon copy.
User	String	User account for SMTP authentication (CheckFlag must be 1?).
PassWord	String	Password for SMTP authentication (CheckFlag must be 1?).
CheckFlag	Integer	Authentication with SMTP (0=No, 1=Yes).
Enable	Integer	Enabled email notification (0=No, 1=Yes).
Interval	Integer	Milliseconds between image updates.

2.7. GetFtp

Gets the FTP settings for motion detection notification.

2.7.1. Parameters

None known.

2.7.2. Response

The response consists of 6 name / value paris in the following order.

Name	Type	Description
FtpServer	String	IP (/ hostname?) of the FTP server.
User	String	Username for FTP authentication.
Pass	String	Password for FTP authentication.
Account	String	Account on FTP server (rarely used?).
UploadPath	String	Path to upload files to.
Enable	Integer	Enable FTP notification (0=No, 1=Yes).
Interval	Integer	Milliseconds between image updates.

2.8. GetUser

Get a list of the available users.

2.8.1. Parameters

None known.

2.8.2. Response

A single user name will be returned on each line of the response (text/plain). I.e.

```
<user1>[CR]
<user2>[CR]
..
..
<userX>[CR]
```

2.9. GetVer

Gets the current software / camera version / date?

2.9.1. Parameters

None known.

2.9.2. Response

A single line (text/plain) containing the version (a date in fact).

2.10. GetLog

Gets the event log

2.10.1. Parameters

None known

2.10.2. Response

The response body contains A single log event on each line separated by [CR]. The content type is text/plain. The format of each record is currently unknown.

```
<event1>[CR]
```

```
<event2>[CR]
..
..
<eventX>[CR
```

2.11. GetIP

Gets various TCP/IP settings.

2.11.1. Parameters

A single parameter is accepted.

Name	Type	Description
Interface	Integer	eth1=Cable, wlan0=Wireless

2.11.2. Response

The response consists of 9 name / value paris in the following order.

Name	Type	Description
CameraName	String	The configured camera name.
DNS0	String	The primary DNS server for manual configuration. (Note this does not appear to contain any DHCP configured DNS servers).
DNS1	String	The secondary DNS server for manual configuration. (Note this does not appear to contain any DHCP configured DNS servers).
DNS2	String	The tertiary DNS server for manual configuration. (Note this does not appear to contain any DHCP configured DNS servers).
Enable	Integer	Interface enabled. 0=No, 1=Yes.
IPWay	String	Interface configuration method. dhcp=DHCP, manually=Manually configured address.
Netmask	String	Dotted netmask for manual configuration.
Gateway	String	IP address of LAN router.
MailOnDhcp	String	This may be similar to the PPPoE facility to email the cameras IP address to someone as soon as it is configured via DHCP. However, some vital parameters such as mail server etc are present in PPPoE settings but not here.
MailSubject	String	See MailOnDhcp above.

2.12. GetHttp

Returns the port numbers on which the Http server is listening.

2.12.1. Parameters

None known.

2.12.2. Response

The response consists of 2 name / value paris in the following order.

Name	Type	Description
Port0	Integer	The first port on which the server listens to. Defaults to 80.
Port1	Integer	The second port on which the server listens to. Defaults to 0 (Disabled?).

2.13. GetCapability

Returns what the camera can do?

2.13.1. Parameters

None known.

2.13.2. Response

The response consists of 7 name / value paris in the following order. We are unsure as to how to interpret these values.

Perhaps it is suspicious that Brightness and Contrast both return values of 1 and neither of the corresponding *ChangeBrightness* or *ChangeConstrast* commands have any effect.,

Name	Type	Description
Brightness	Integer	Unknown.
Contrast	Integer	Unknown.
Saturation	Integer	Unknown.
Hue	Integer	Unknown.
Sharpness	Integer	Unknown.
Resolution	Integer	Unknown.
Quality	Integer	Unknown.

2.14. SetHttp

Sets the port numbers on which the Http server is listening.

2.14.1. Parameters

Two parameters are accepted..

Name	Type	Description
Port0	Integer	The first port on which the server listens to. Defaults to 80.
Port1	Integer	The second port on which the server listens to. Defaults to 0 (Disabled?).

2.14.2. Response

Empty body.

2.15. ChangeResolution

Sets the image resolution.

2.15.1. Parameters

A single parameter is accepted.

Name	Type	Description
ResType	Integer	0=176x144,1=352x288,2=320x240, 3=640x480

2.15.2. Response

Empty body.

2.16. ChangeCompressRatio

Sets the compression ratio ('Quality' as presented in the web UI).

2.16.1. Parameters

A single parameter is accepted.

Name	Type	Description
Ratio	Integer	Quality. 0=Lowest,4=Highest

2.16.2. Response

Empty body.

2.17. ChangeHue

Change the hue of the camera image.

2.17.1. Parameters

A single parameter is accepted.

Name	Type	Description
Hue	Integer	Signed value. (0 - 255, -1 is the default).

2.17.2. Response

Empty body.

2.18. ChangeSaturation

Change the saturation of the camera image.

2.18.1. Parameters

A single parameter is accepted.

Name	Type	Description
Saturation	Integer	Signed value. (0 - 255, -1 is the default).

2.18.2. Response

Empty body.

2.19. ChangeSharpness

Change the sharpness of the camera image.

2.19.1. Parameters

A single parameter is accepted.

Name	Type	Description
Sharpness	Integer	Signed value. (0 - 255, -1 is the default).

2.19.2. Response

Empty body.

2.20. ChangeBrightness

Change the brightness of the camera image. Not sure this does anything, or I have not identified the proper parameters.

2.20.1. Parameters

None known.

2.20.2. Response

Empty body.

2.21. ChangeContrast

Change the contrast of the camera image. Not sure this does anything, or I have not identified the proper parameters.

2.21.1. Parameters

None known.

2.21.2. Response

Empty body.

2.22. Reboot

Reboots the camera.

2.22.1. Parameters

A single parameter is accepted..

Name	Type	Description
RebootNow	Integer	Reboot immediately (0=No, 1=Yes).

2.22.2. Response

Empty body.

2.23. SetFactoryDefault

Resets the camera to its factory defaults. Should be followed by a Reboot command?

2.23.1. Parameters

None known.

2.23.2. Response

Empty body.

2.24. PPPoE

Gets (only?) various PPPoE settings. This will be useful when the camera is connected *directly* to some kind of cable / DSL modem and your ISP requires authentication using PPPoE. There is also a feature which enables the camera to send an email upon connection to a configured address containing the cameras IP address - the rest of the values detail SMTP server etc.

2.24.1. Parameters

A single parameter is accepted.

Name	Type	Description
Action	String	Action to take. (GetSetting=Show current settings)

2.24.2. Response

The response consists of 13 name / value paris in the following order.

Name	Type	Description
ConnectOnBoot	Integer	Connect using PPPoE upon camera boot (0=No, 1=Yes).
User	String	The username to use for PPPoE authentication.
Pass	String	The password to use for PPPoE authentication.
Mail	Integer	Send an email upon IP address assignment (0=No, 1=Yes).
MailServer	String	IP address / Hostname of SMTP server to use for mail notification of newly DHCP configured IP address.
Sender	String	Sender email address for mail notification of address.
Subject	String	Subject of email for mail notification of address.

Name	Type	Description
Receiver	String	Receiver email address for mail notification of address.
Receiver_Cc	String	Carbon copy receiver email address for mail notification of address.
Receiver_Bcc	String	Blind carbon copy receiver email address for mail notification of address.
MailUser	String	User to authentication SMTP connection.
MailPassword	String	Password to authenticate SMTP connection.
CheckFlag	Integer	Unknown. (0=Yes?, 1=No?)