OVERVIEW

This document specifies the external RTSP-based application programming interface of the PLANET cameras with firmware version 3.00 and above.

Product and firmware versions

The support for the RTSP API is product and firmware dependent. Please refer to the Release Notes for the actual product for compliance information.

REFERENCES

- RTSP protocol
    - Real Time Streaming Protocol - RFC 2326
- SDP protocol
    - Session Description Protocol - RFC 2327
- HTTP protocol
    - Hypertext Transfer Protocol -- HTTP/1.0
- External application programming interfaces (Client side)
    - PLANET Video API, HTTP
    - PLANET Video API, RTSP
    -
    - PLANET Video Parameter specification

RTSP API

The RTSP URL is :

```
 * rtsp://<server name>/mpeg4/media.3gp
for 3GPP (QQVGA)

 * rtsp://<server name>/mjpg/1/video.mjpg
for JPEG (default)

 * rtsp://<server name>/mjpg/1/video.mjpg?resolution=vga
for JPEG @ VGA

 * rtsp://<server name>/mjpg/1/video.mjpg?resolution=cif
for JPEG @ CIF
```

```
 * rtsp://<server name>/mjpg/1/video.mjpg?resolution=qcif
for JPEG @ QCIF

 * rtsp://<server
name>/mjpg/1/video.mjpg?resolution=1280x1024 for JPEG @
1280x1024

 * rtsp://<server name>/mpeg4/1/media.amp
for MPEG4 (default)

 * rtsp://<server name>/mpeg4/1/media.amp?resolution=vga
for MPEG4 @ VGA

 * rtsp://<server name>/mpeg4/1/media.amp?resolution=cif
for MPEG4 @ CIF

 * rtsp://<server name>/mpeg4/1/media.amp?resolution=qcjf
for MPEG4 @ QCIF

 * rtsp://<server
name>/mpeg4/1/media.amp?resolution=1024x768  for MPEG4 @
1024x768
```

where <server name> is the host name or IP address of the server.
The DESCRIBE, SETUP, OPTIONS, PLAY, PAUSE and TEARDOWN methods are supported. The RTSP protocol is described in RFC 2326. Request syntax:

```
COMMAND URI RTSP/1.0<CRLF>

Headerfield1: val1<CRLF>

Headerfield2: val2<CRLF>

...

<CRLF>
```

Response syntax:

```
RTSP/1.0 ResultCode ResultString<CRLF>
```

```
Headerfield3: val3<CRLF>

Headerfield4: val4<CRLF>

...

<CRLF>
```

The following header fields are accepted by all commands. Other header fields are silently ignored (unless stated otherwise in the sections below).

Field	Description
Authorization	Authorization information from the client.
CSeq	Request sequence number.
Session	Session identifier (returned by server in SETUP response).
Content-Length	Length of content.

The following header fields can be generated for all responses by the RTSP server:

Field	Description
CSec	Response sequence number (matches the sequence number of the request).
Session	Session identifier.
WWW-Authenticate	Authentication from client requested.

RTSP DESCRIBE

- The DESCRIBE command returns the SDP (RFC 2327) description for the URI.
- The DESCRIBE command accepts the following additional header fields:

Field	Description
Accept	List of content types that client supports (application/sdp is the only supported type).

The DESCRIBE command generates the following additional header fields:

Field	Description

Content-Type	Type of content (application/sdp).
Content-Length	Length of SDP description.
Content-Base	If relative URLs are used in the SDP description, then this is the base URL.

Example:

```
DESCRIBE /mpeg4/1/media.amp RTSP/1.0

CSec: 1

Accept: application/sdp

Authorization: Basic cm9vdDpwYXNz
```

Response example:

```
RTSP/1.0 200 OK

CSec: 1

Content-Base: rtsp://x.y.z.w:554/mpeg4/quad/media.amp

Content-Type: application/sdp

Content-Length: 680



v=0

o=- 1109162014219182 1109162014219192 IN IP4 x.y.z.w

s=Media Presentation

e=NONE

c=IN IP4 0.0.0.0

b=AS:8000

t=0 0
```

```
a=control:*

a=range:npt=now-

a=mpeg4-iod:
"data:application/mpeg4-iod;base64,AoDUAE8BAf/1AQOAbwABQFBk
YXRhOmFwcGxpY2

F0aW9uL21wZWc0LW9kLWF1O2Jhc2U2NCxBUjBCR3dVZkF4Y0F5U1FBW1FRT
klCRUVrK0FBZWhJQUFIb1NBQV1C

QkE9PQQNAQUABAAAAAAAAAAAAAYJAQAAAAAAAAAAAzoAAkA2ZGF0YTphcHB
saWNhdG1vbi9tcGVnNC1iaWZzLW

F1O2Jhc2U2NCx3QkFTWVFTSVVFVUZQd0E9BBICDQAAAgAAAAAAAAABQMAA
EAGCQEAAAAAAAAAAA=="

m=video 0 RTP/AVP 96

b=AS:8000

a=control:trackID=1

a=rtpmap:96 MP4V-ES/90000

a=fmtp:96 profile-level-id=245;
config=000001B0F5000001B509000001000000012008D495880325

0B042414440F;

a=mpeg4-esid:201
```

RTSP OPTIONS

The OPTIONS command returns a list of supported RTSP commands. Example:

```
OPTIONS * RTSP/1.0

CSec: 2
```

Response example:

```
RTSP/1.0 200 OK
```

```
CSec: 2

Public: DESCRIBE, PAUSE, PLAY, SETUP, TEARDOWN
```

RTSP SETUP

The SETUP command configures the delivery method for the data. The SETUP command requires and generates the following additional header field:

Field	Description
Transport	Specifies how the data stream is transported.
	Supported variants:
	RTP/AVP;unicast;client_port=port1-port2
	RTP/AVP;multicast;client_port=port1-port2
	RTP/AVP/TCP;unicast

The response returns a session identifier that should be used with stream control commands to the server (PLAY, PAUSE, TEARDOWN). If the Session header includes a timeout parameter, then the session needs to be kept alive. This can be done by sending RTSP requests to the server containing the session identifier (e.g. OPTIONS) within the specified timeout time or through the use of RTCP. The RTSP server does not support reconfiguring of the transport parameters. Example:

```
SETUP /mpeg4/1/media.amp/trackID=1 RTSP/1.0

CSeq: 3

Transport: RTP/AVP;unicast;client_port=20000-20001
```

Response example:

```
RTSP/1.0 200 OK

CSeq: 3

Session: 1442774578;timeout=60

Transport:
RTP/AVP;unicast;mode=play;server_port=49000-49001;client_port=20000-20001
```

RTSP PLAY

The PLAY command starts (or restarts if paused) the data delivery to the client. The PLAY command generates the following additional header fields:

Field	Description
Range	Specifies the range of time being played. Since only live streams are used, the specified time will always begin now and have no stop time.
RTP-Info	Information about the RTP stream. More specifically, it includes the next RTP sequence number that will be used.

Example:

```
PLAY /mpeg4/1/media.amp RTSP/1.0

CSeq: 4

Session: 1442774578
```

Response example:

```
RTSP/1.0 200 OK

CSeq: 4

Session: 1442774578

Range: npt=now-

RTP-Info: url=trackID=1;seq=2735
```

RTSP PAUSE

The PAUSE command pauses the data delivery from the server. Example:

```
PAUSE /mpeg4/1/media.amp RTSP/1.0

CSeq: 5

Session: 1442774578
```

Response example:

```
RTSP/1.0 200 OK

CSeq: 5

Session: 1442774578
```

RTSP TEARDOWN

The TEARDOWN command terminates the data delivery from the server. Example:

```
TEARDOWN /mpeg4/1/media.amp RTSP/1.0

CSeq: 6

Session: 1442774578
```

Response example:

```
RTSP/1.0 200 OK

CSeq: 6

Session: 1442774578
```

RTSP over HTTP

It is possible to tunnel RTSP through HTTP to get through firewalls etc. This is achieved by setting up two sessions, one GET (for command replies and stream data) and one POST (for commands). RTSP commands sent on the POST connection are base64 encoded, but replies on the GET connection are plain text. To bind the two sessions together the server needs a unique ID (conveyed in the x-sessioncookie header). The GET and POST request are accepted both on the HTTP port (default 80) and the RTSP server port (default 554). The RTSP/HTTP URL is: http//<server name>/mpeg4/1/media.amp. Supported methods are GET and POST.
GET example:

```
GET /mpeg4/1/media.amp HTTP/1.0

x-sessioncookie: 944345808
```

Response example:

```
HTTP/1.0 200 OK

Content-Type: application/x-rtsp-tunnelled
```

Note that the returned Content-Length is an arbitrary number and does not indicate the size of any data.
POST example:

```
POST /mpeg4/1/media.amp HTTP/1.0

x-sessioncookie: 944345808

Content-Length: 32767

Content-Type: application/x-rtsp-tunnelled
```

Note that the x-sessioncookie present in the POST content is the same as the x-sessioncookie in the corresponding GET request. Also note that the POST request must be sent on a separate connection to the server (i.e. it cannot be sent on the same connection as the GET request). No response is sent for the POST request.
Command example (base 64 encoded PLAY on the POST connection):

```
ExBWSAvbXBlZzQvMS92aWR1by5tcGcgU1RTUC8xLjANCkNTZXE6IDQNC1Vz
ZXItQWdlbnQ6

IEF4aXMgQU1DDQpTZXNzaW9uOiAxMzY5OTE3NzEyDQoNCg==
```

Which corresponds to this command:

```
PLAY /mpeg4/1/media.amp RTSP/1.0

CSeq: 4

Session: 1369917712
```

Command response (plain-text on the GET connection):

```
RTSP/1.0 200 OK

CSeq: 4
```

```
Session: 1369917712
```