**DOCUMENT FOR RTSP SERVER**

**MAIN FILE** – rtsp\_server.c

**Client handling and request handing** – rtspservice.c

**TCP and Scheduler Utilities** – rtsputils.c

**Rtp packet handling** – rtputils.c

**Handling ring buffer** – ringfifo.c

**RTSP URL THAT IS ACCPTED BY THIS RTSP SERVER CODE**

**FOR MAIN STREAM**

**Channel number can be changed by Set\_Channel\_RTSP in rtsp\_server.c**

rtsp://username:password@IP:PORT/cam/realmonitor?channel=1&subtype=0 – if authentication is on

rtsp://IP:PORT/cam/realmonitor?channel=1&subtype=0 – if authentication is off

**FOR SUB STREAM**

**Channel number can be changed by Set\_Channel\_RTSP in rtsp\_server.c**

rtsp://username:password@IP:PORT/cam/realmonitor?channel=1&subtype=1 – if authentication is on

rtsp://IP:PORT/cam/realmonitor?channel=1&subtype=1 – if authentication is off

**NOTE – PORT for rtsp is 554 (default)**

**MAIN FILE** – rtsp\_server.c

* **void Set\_Channel\_RTSP(unsigned int channel)**
* This function is used to set the RTSP channel number (as through channel number NVR connects to camera. So, you must set the channel number before calling rtsp\_server\_init function).
* Default channel number -- 1
* **unsigned int Get\_Channel\_RTSP()**
* This function is used to get the rtsp channel number. You can check if the channel number has been saved by Set\_Channel\_RTSP.
* **int Send\_Video\_RTSP(unsigned char \*pstStream,unsigned int length,unsigned int video\_type,unsigned int stream\_type)**
* This function is used to send the rtsp video data to the ring buffer so that we can send the buffer to the client.
* This function is to be called where video polling is being done.
* stream\_type - MAIN STREAM or SUB STREAM
* video\_type – H264 or H265 video type
* **int PutPCMDataToBuffer(unsigned char \*pstStream,unsigned int length,unsigned int audio\_type)**
* This function is used to send the rtsp audio data to the ring buffer so that we can send the buffer to the client.
* This function is to be called where audio polling is being done.
* audio\_type – PCM LINEAR , PCM A LAW or PCM U LAW audio type
* **int Set\_RTSP\_Basic\_Authorization(char\* username,char\*password)**
* This function is used to set RTSP authorization.
* You must give the username and password to secure the RTSP.
* You must call this function before calling rtsp\_server\_init function
* If you will not call this function the RTSP server will run in no authentication mode
* **int** **rtsp\_server\_init()**
* This function initializes the rtsp server.
* It creates two threads (one for handling RTSP requests and other for sending and receiving data buffers for clients )
* **void RTSP\_Server\_DeInit()**
* This fuction free all the buffers for RTSP server and stops RTSP server.
* **void\* rtsp\_server\_connect()**
* This is the thread which is used to handle rtsp client’s connection and requests

**AUTHENTICATION**

We use basic authentication for RTSP. Rtsp client sends username and password with URL. Then the client software encodes these username and password in base64 format and sends to the server . Server decodes the base64 string and compares the username and password. If it matches with the username and password stored in the RTSP server then it will respond RTSP 2.0 OK. Otherwise it will respond unauthorized to RTSP Server.

In this code , In rtspservice.c (function –RTSP\_State\_Machine) we decode this username and password and then compares with the username and password and according to the result we send the reply to RTSP Client.

**RING BUFFER FOR VIDEO AND AUDIO**

Ring buffer is filled by functions **PutPCMDataToBuffer** and **Send\_Video\_RTSP** which are called by audio polling and video polling function. Then the Scheduler thread (schedule\_do function) willcheck the data in the ring buffer and if there is any data then it will check the stream type (MAIN\_STREAM, SUB\_STREAM or AUDIO), then it sends to the RTSP clients.

**PROTOCOL DIRECTIVES**

RTSP Server supports both UDP(unicast) and TCP through RTP.

RTSP client sends the Requests(OPTIONS, DESCRIBE,SETUP and PLAY) through TCP. When RTSP clients sends the SETUP packet it also has the way to send the video and audio buffer(TCP or UDP).

According to the request Server sends the reply to the client. Then, the server sends the audio and video buffer according to the request of the client