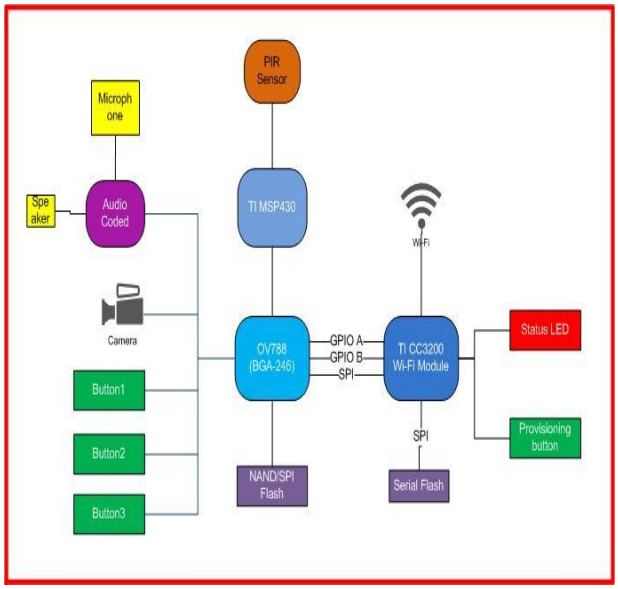
TI Designs: Reference Designs

CC3200-OV788 Video Doorbell – Design Guide

Design Features

- The design combines TI and OmniVision's leading wireless and AV technologies to bring live streaming capabilities of audio and video data over Wi-Fi.
- Supports capture and streaming of:
  - Video - 720p @ 15FPS
  - Audio – PCM, 16bps @ 11025Hz
- Supports RTP/RTSP protocols – Compatible with media-players supporting these protocols
- SimpleLink™ Wi-Fi connectivity over 802.11 b/g/n networks from any smart phone, tablet, or computer over local network
- Inbuilt provisioning for easy commissioning of the device to a Wi-Fi network
- Advanced Low-Power Modes

Block Diagram



Design Resources

- CC3200
- OV788
- OVI9712 Camera Sensor

Featured Applications

- Wireless Video Doorbells
- Home/Baby Monitoring
- Remote Surveillance
- Wireless Announcement Systems

TI E2E™ Community

Ask the SimpleLink™ Wi-Fi® Experts

Copyright © 2016, Texas Instruments Incorporated

Version 1.0
August 29, 2016

CC3200 Video Doorbell – Design Guide

(Texas Instruments NDA restrictions applicable)

Contents

Copyright © 2016, Texas Instruments Incorporated

Copyright © 2016, Texas Instruments Incorporated

Introduction

This CC3200-OV788 based Video Doorbell design combines TI and OmniVision's leading wireless and AV technologies to bring live streaming capabilities of audio and video data over Wi-Fi. The exponential increase in the demand for Wi-Fi-enabled battery operated cameras, for applications ranging from video doorbells to surveillance devices, has fueled the need for reference designs of this kind, consisting of all the required building blocks to help developers enable the video/audio functionalities into their end applications faster.

This design provides an integrated solution showcasing the CC3200's ability to provide full system solution for audio-video streaming applications. On boot-up, the application initializes the OV788 subsystem, configures CC3200 to connect to a Wi-Fi network, opens an RTSP server and waits for the RTSP clients, like media-player applications, to connect and request for live streams.

Figure-1 below shows a high level block diagram of all the various components/modules of this reference design. System shall be powered up using a 3.6V supply.

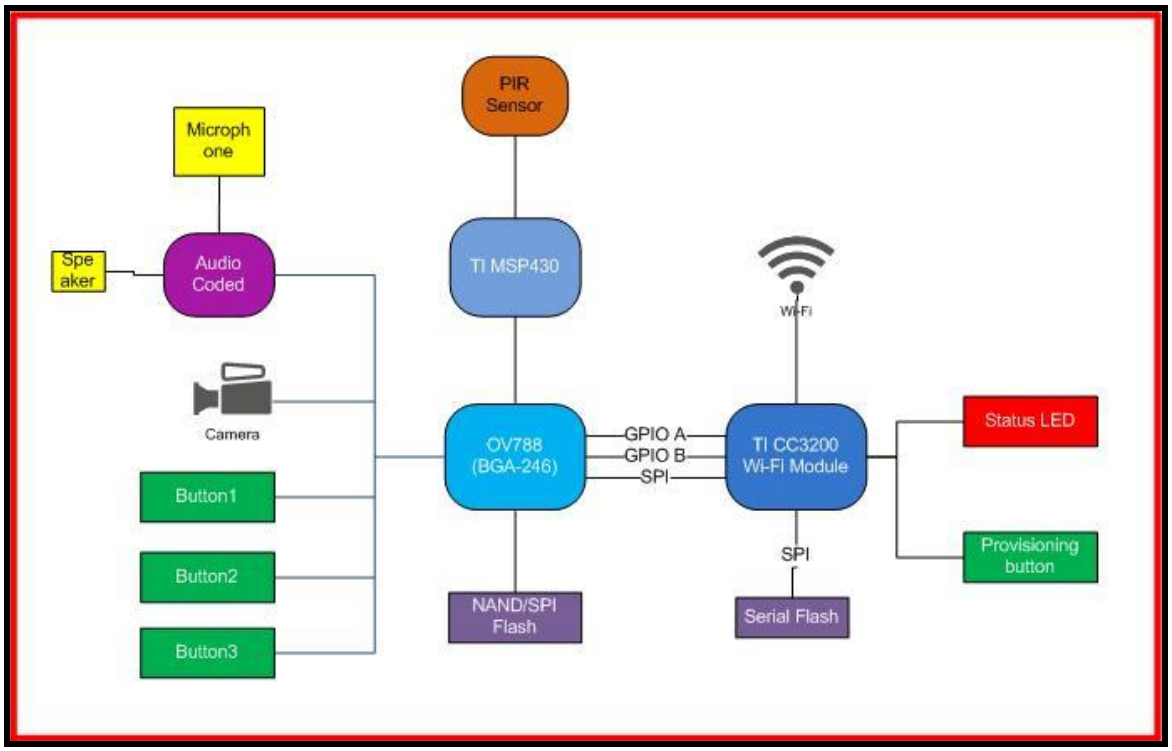


Figure-1: CC3120 Video Camera Block Diagram

Copyright © 2016, Texas Instruments Incorporated

Getting Started

Prerequisites

- 1x CC3200-MODULE Rev 2.0
- 1x OV788 Reference Design Board Rev 3.0
- 1x OV9712 CMOS Camera Sensor w/ optic lens
- 1x Apple's iPhone/iPad or Android Smartphone
- 1x 802.11 b/g/n Wi-Fi Access Point
- 1x Micro-USB cable
- For programming/logging:
    - 1x CC3200-LAUNCHXL Rev 4.1
    - 1x PC running Microsoft Windows® 7

Installation of Tools and Software

- Download and install the following software components from ti.com to the Windows-PC. This guide assumes the components are downloaded to location 'C:\TI'
    - CC3200SDK v1.2.0
    - CC3200SDK-PROVISIONING v1.0.0.0 (To be installed into the above SDK)
    - CC3200SDK-SERVICEPACK v1.0.1.6-2.6.0.5

  For installation/setup details of the SDK, refer the CC3200-SDK's Getting Started Guide.
- Download the CC3200 Video Doorbell package 'cc3200_video_doorbell_v01' from TI's shared location to the Windows-PC and copy 'cc3200_video_doorbell_v01\cc3200-sdk*' to 'C:\TI\CC3200SDK_1.2.0\cc3200-sdk\'. This action will copy the following folders:
    - 'cc3200-sdk\docs' to 'C:\TI\CC3200SDK_1.2.0\cc3200-sdk\docs'

Copyright © 2016, Texas Instruments Incorporated

- ‘cc3200-sdk\example\video_camera’ to ‘C:\TI\CC3200SDK_1.2.0\cc3200-sdk\example’
- ‘cc3200-sdk\netapps\rtp_rtcp’ and ‘cc3200-sdk\netapps\rtsp’ to ‘C:\TI\CC3200SDK_1.2.0\cc3200-sdk\netapps’
- ‘cc3200-sdk\third_party\ov_sif_inerface’ and ‘cc3200-sdk\third_party\ov788_firmware’ to ‘C:\TI\CC3200SDK_1.2.0\cc3200-sdk\third_party’

- Install the provisioning application on iOS/Android Smartphone
- Download the ‘Uniflash’ tool from ‘ti.com’ and install on the Windows PC
- Download ‘VLC v2.7’ & ‘Discovery’ iOS application from Apple’s App-Store and install it on the iOS device. Alternatively, download ‘RTSP Player v4.4.0’ & ‘Bonjour Browser’ Android application from Android’s Play-Store and install it on the Android Smartphone

Updating the device software

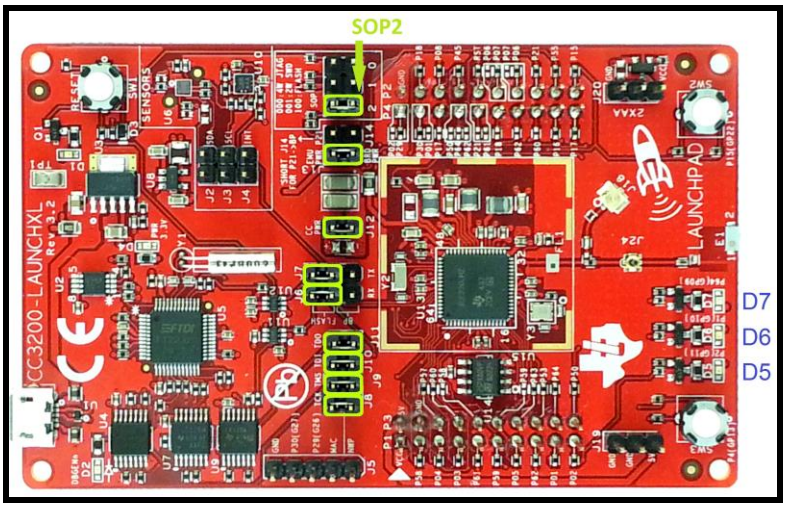
If the service-pack, application binaries and OV firmware files are already programmed on CC3200-MOD device, proceed to Section: ‘Setting up the demonstration’.

Programming the CC3200-MOD device

- Mount CC3200-MODULE board on OV788 board as shown in the picture – Ensure J1 of both the boards are aligned with each other

Copyright © 2016, Texas Instruments Incorporated

- Assuming the jumpers on the CC3200-LAUNCHPAD are connected as shown below, remove J6, J7, J8, J9, J10, J11, J12 and J15



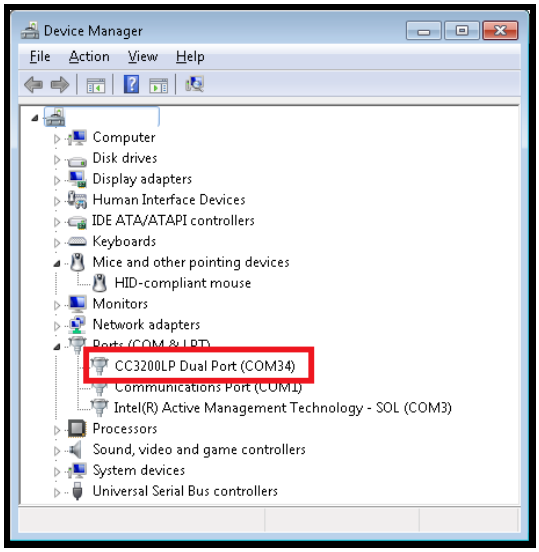
- Connect UART Tx line of CC3200-MODULE board w/ Pin-1 of J7 on CC3200-LAUNCHXL as shown below [1]

- Connect UART Rx line of CC3200-MODULE board w/ Pin-1 of J6 on CC3200-LAUNCHXL as shown below [2]

- Connect SOP2 line of CC3200-MODULE board w/ VCC or Pin-1 of J15 on CC3200-LAUNCHXL as shown below [3]

Copyright © 2016, Texas Instruments Incorporated

- Connect J1 port of CC3200-LAUNCHXL to the PC using a micro-USB cable. The device should be visible in the PC's 'Device Manager' as shown below, and a COM port shall be enumerated.



- Power up the OV788/CC3200-MODULE setup using the 'Battery Connector' as shown above

- Launch the 'Uniflash' tool and perform the following operations

    - Follow 'Format' button to format the serial flash on CC3200-MODULE board.

    - Follow 'Service Pack Programming' button to program service-pack 'v1.0.1.6-2.6.0.5' on CC3200-MODULE board

    - Add file from 'Operation → Add File' menu and name it '/user/ovt_firmware.bin'. For the URL, choose 'dsif_slave.bin' from 'cc3200-sdk\third_party\ov788_firmware'. Check 'Erase', 'Update' and 'Verify'

    - For '/sys/mcuimg.bin', choose 'video_camera.bin' from 'cc3200-sdk\example\video_camera\ewarm\Release_LegacyProvisioning\Exe' or 'cc3200-sdk\example\video_camera\ccs\Release_LegacyProvisioning'. Check 'Erase', 'Update' and 'Verify'

    - Follow 'Program' button to program the OV firmware and application binary on CC3200-MODULE board.

Note: CC3200-MODULE board doesn't have a 'RESET' button – Pull out the power and plug it back in when a reset is required during the programming process.

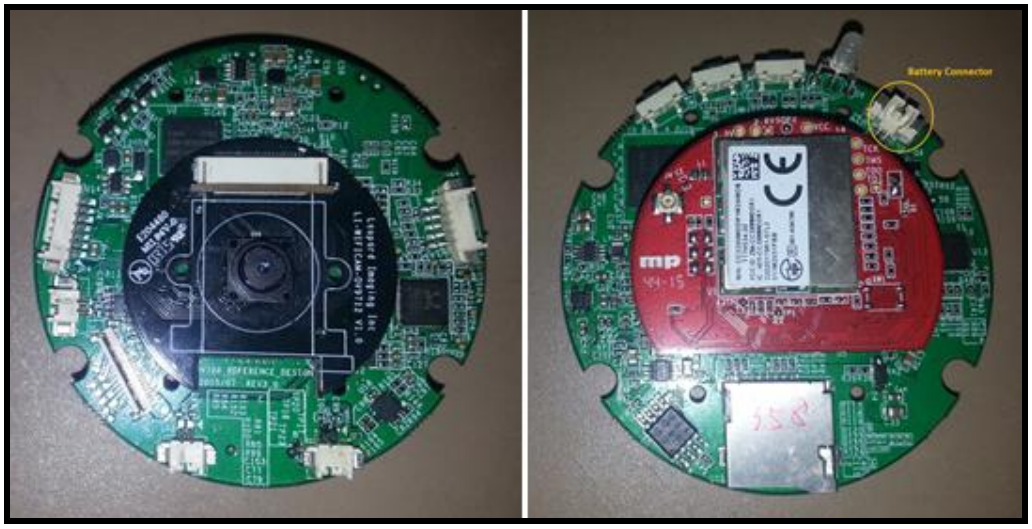
Copyright © 2016, Texas Instruments Incorporated

Note: If new Gen-1 provisioning is to be used, 'simplelink_extlib/provisioninglib/ewarm/Release/Exe/provisioninglib.a' shall be build w/ 'SL_PLATFORM_MULTI_THREADED' defined and path set to '$PROJ_DIR$/../../../oslib/' path shall be included. Also, 'simplelink.a' shall be rebuilt.

Setting up the demonstration

Board Configuration

- Mount CC3200-MODULE board on OV788 board as shown in the picture – Ensure J1 of both the boards are aligned with each other

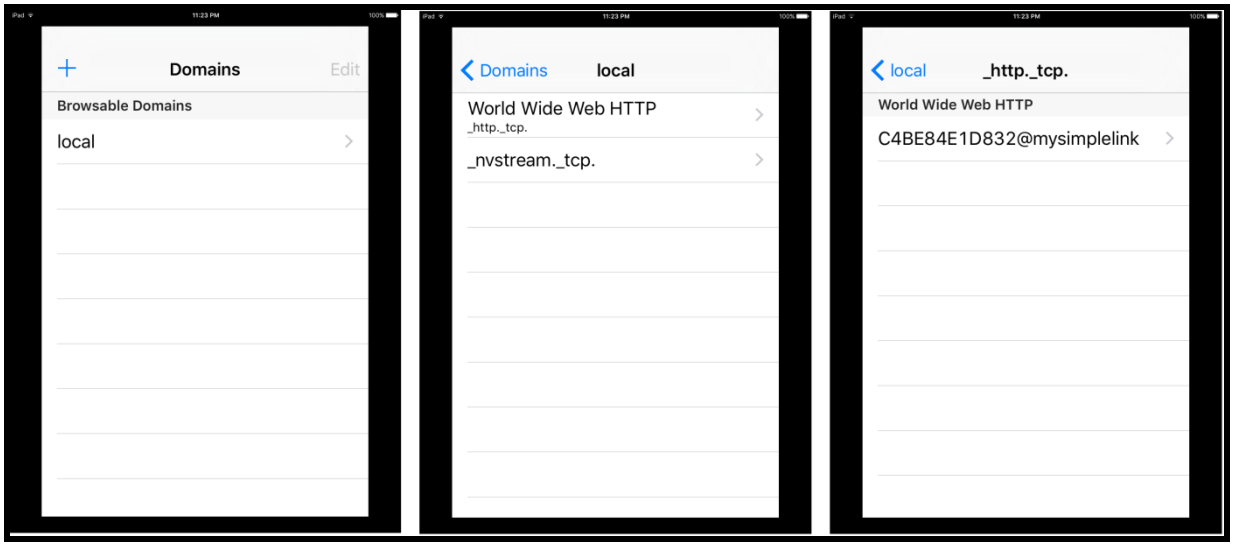


- Power up the OV788/CC3200-MODULE setup using the 'Battery Connector' shown below. On powering up:
    - The application shall wait for ~10 seconds to establish a connection w/ a known AP. This state is indicated by a blinking 'Red LED' on CC3200-MOCULE board.
    - If the device fails to connect w/ a known AP or if there are no saved profiles, the device will enter the provisioning mode. This state is indicated by solid 'Red LED'. Use the 'Simplelink Starter Pro' or 'Wi-Fi Starter' Smartphone application to configure the device.
    - The 'Red LED' will turn off if the connection is successfully established.
    - The application then initializes the OV788 board, configures the audio and video modules, starts the RTSP server and listens for client connections

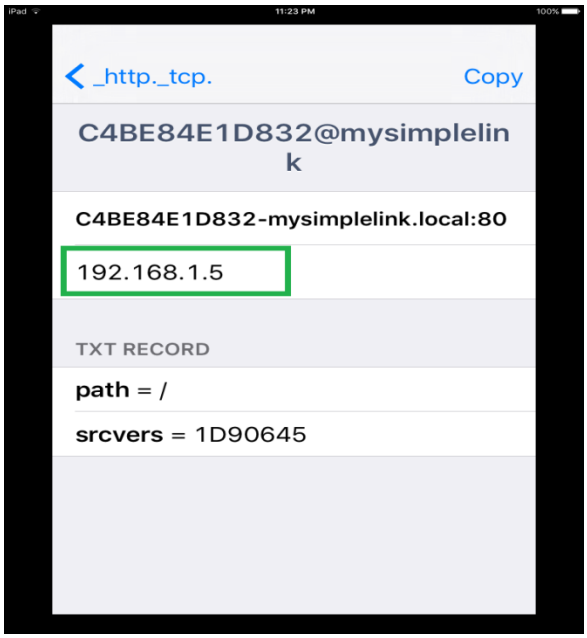
Copyright © 2016, Texas Instruments Incorporated

AV Streaming with iPad/iPhone

- Ensure that the applications listed in Section:2.2 are installed on iPad/iPhone
- Connect the Smartphone to the same network that the demo device is connected to. Also, connect a headphone using the 3.5mm jack
- Open the "Discovery" application and navigate to 'local → World Wide Web HTTP → xxxxxxxxxxx@mysimplelink'

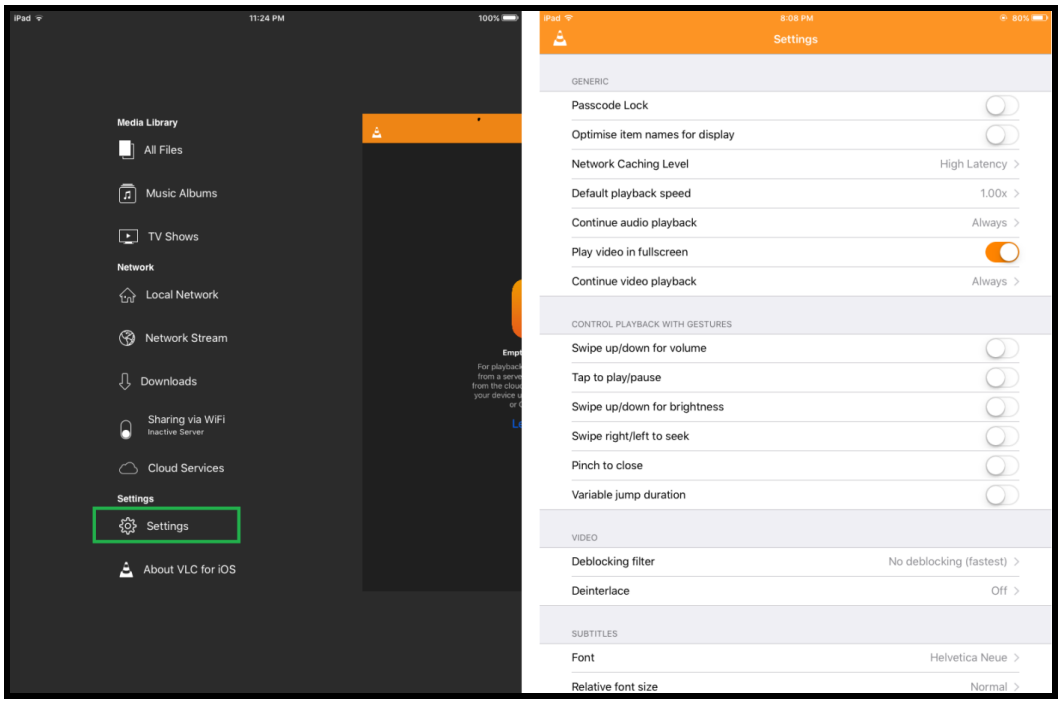


- Note down the IP address of CC3200

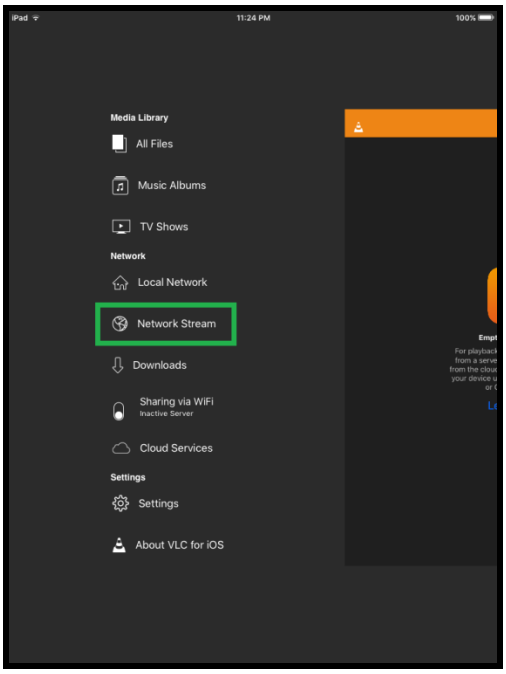


Copyright © 2016, Texas Instruments Incorporated

- Open the VLC application and navigate to “Settings” tab and use the settings shows in following image.

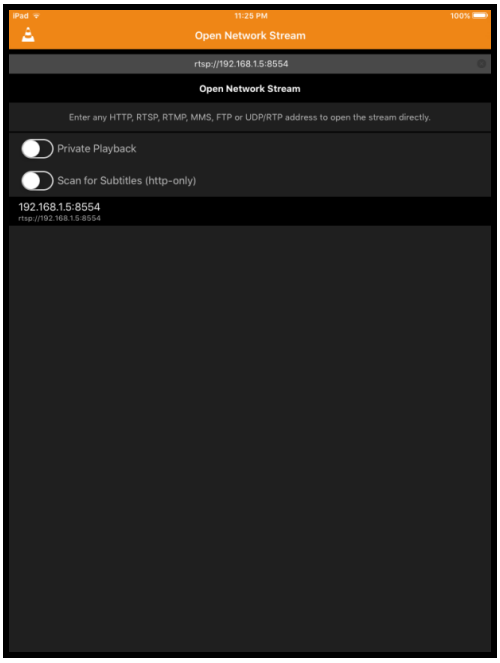


- Navigate to ‘Network Stream’ tab.



Copyright © 2016, Texas Instruments Incorporated

- Enter ‘rtsp://<demo-device-ip>:8554’ and press ‘Open Network Stream’.



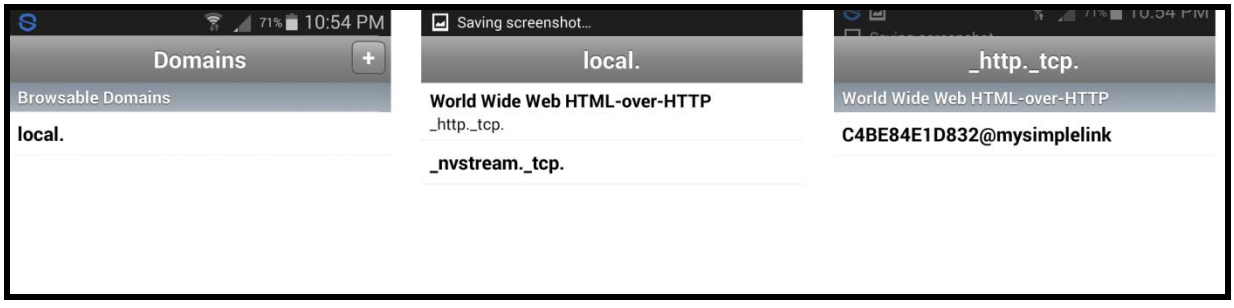
- VLC application will connect to the device and start playing the live video and audio streams

AV Streaming with Android Smartphone

- Ensure that the applications listed in Section:2.2 are installed on the Android Smartphone

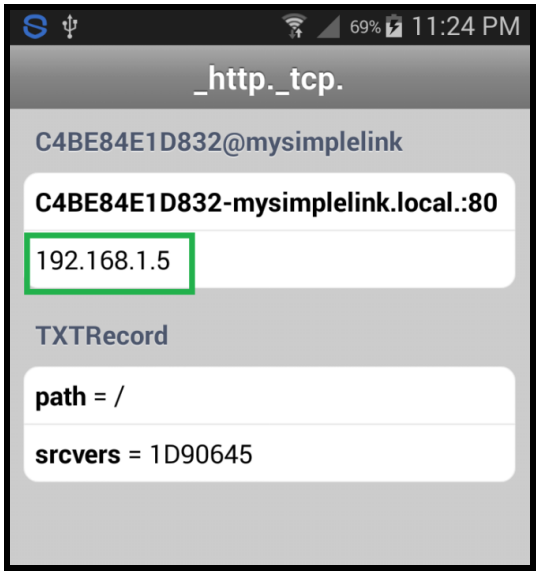
- Connect the Smartphone to the same network that the demo device is connected to. Also, connect a headphone using the 3.5mm jack

- Open the ‘Bonjour Browser’ application and navigate to ‘local → World Wide Web HTML-over-HTTP → xxxxxxxxxxxx@mysimplelink’

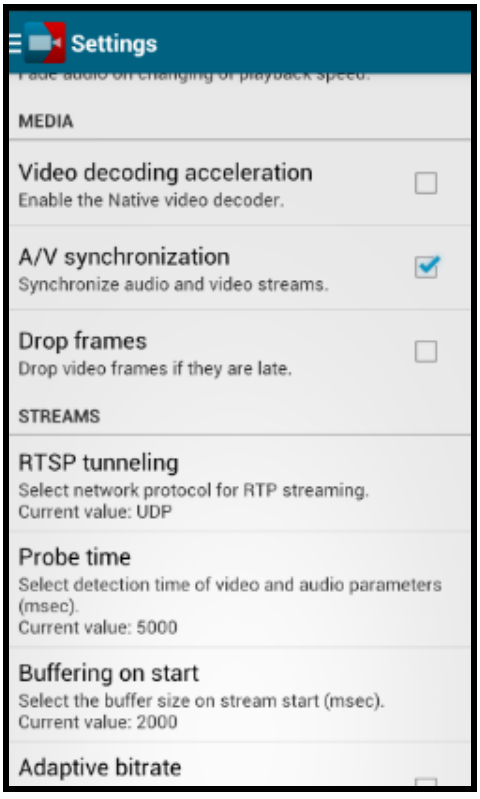


Copyright © 2016, Texas Instruments Incorporated

- Note down the IP address of CC3200

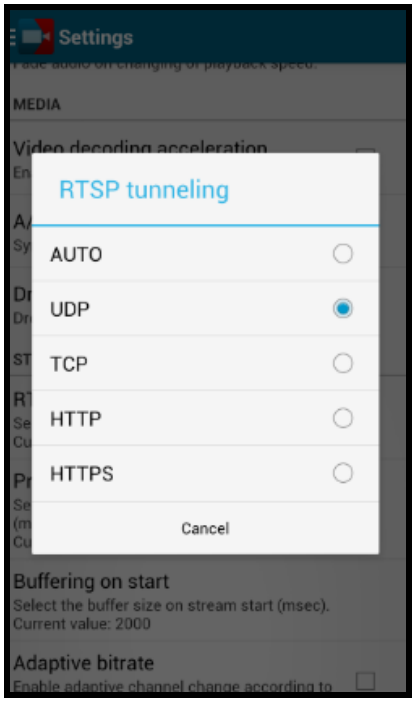


- Open the 'RTSP Player' application and navigate to "Settings" tab and uncheck 'Video decoding acceleration' and 'Drop frames' under 'MEDIA'.

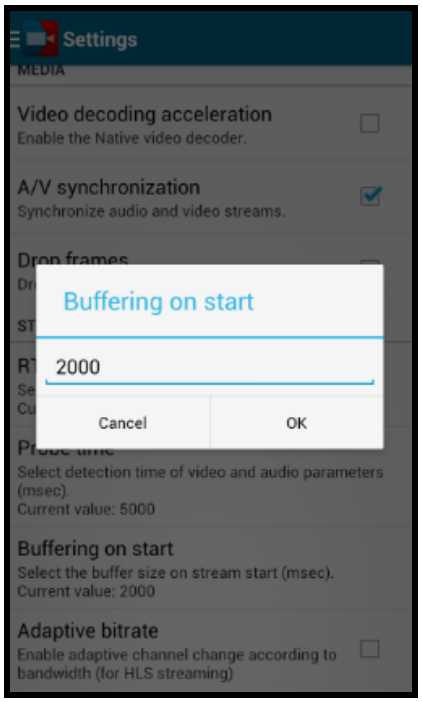


Copyright © 2016, Texas Instruments Incorporated

- Select ‘UDP’ under ‘STREAMS → RTSP tunneling’ as shown below

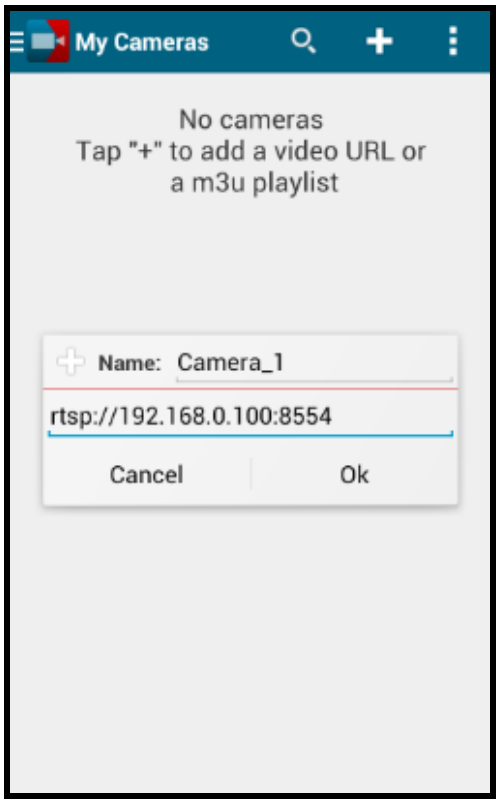


- Enter ‘2000’ under ‘STREAMS → Buffering on start’



Copyright © 2016, Texas Instruments Incorporated

- Add a ‘Camera’ by pressing on the ‘+’ button as shown below – Enter a ‘Name:’ and ‘rtsp://<demo-device-ip>:8554’, and press ‘Ok’



- ‘RTSP Player’ application will connect to the device and start playing the live video and audio streams

Detailed Design Description

Hardware Description

The CC3200 Video camera hardware is divided into two sections.

- Wi-Fi + Host sub section.
- Compressing and Sensor sub section – OVI

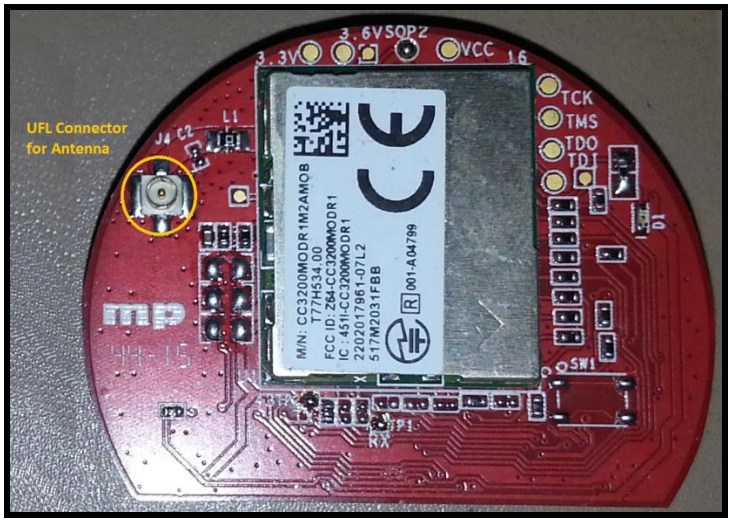
Copyright © 2016, Texas Instruments Incorporated

Wi-Fi + Host Sub section

CC3200 Wi-Fi board is compact designed board for video camera which enables this solution to be used in compact designs. SN74AVC4T245PW level shifter is used to interface the 3.3V CC3200 signals with 1.8V signals of OV788

Antenna Connection

Antenna is not available CC3200 Wi-Fi board. Antenna can be attached using the UFL connector as shown in below image.



Programming Interface

There are TP available on the CC3200 Wi-Fi board for programming and debugging. These interfaces are showcased in below image.

Copyright © 2016, Texas Instruments Incorporated

Compressing and Sensor sub section - OVI

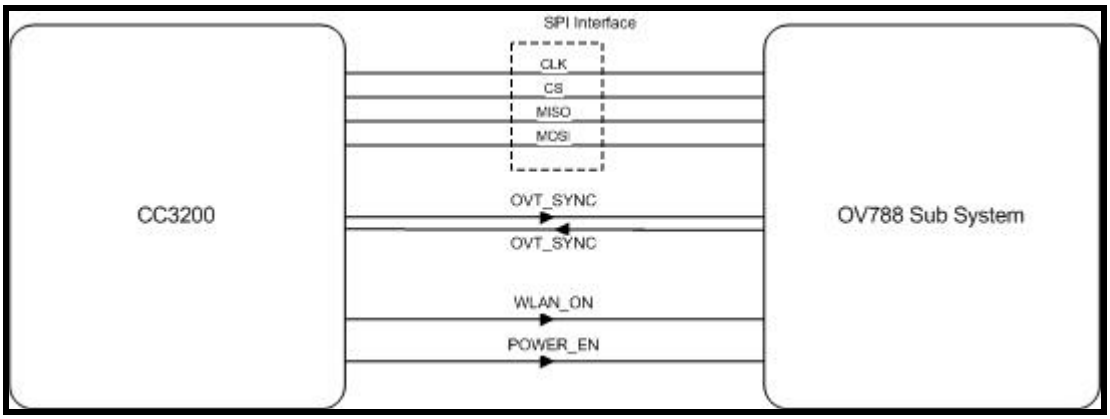
OmniVision's OV9712 CMOS image sensor and OV788 video compressing module is used for HD video streaming. To run the demo OV788 should be configured in boot 1101 mode.

Contact person for detailed information related to OV788 hardware

David Ho, OmniVision Technologies

dho@ovt.com<mailto:dho@ovt.com

Connection Diagram



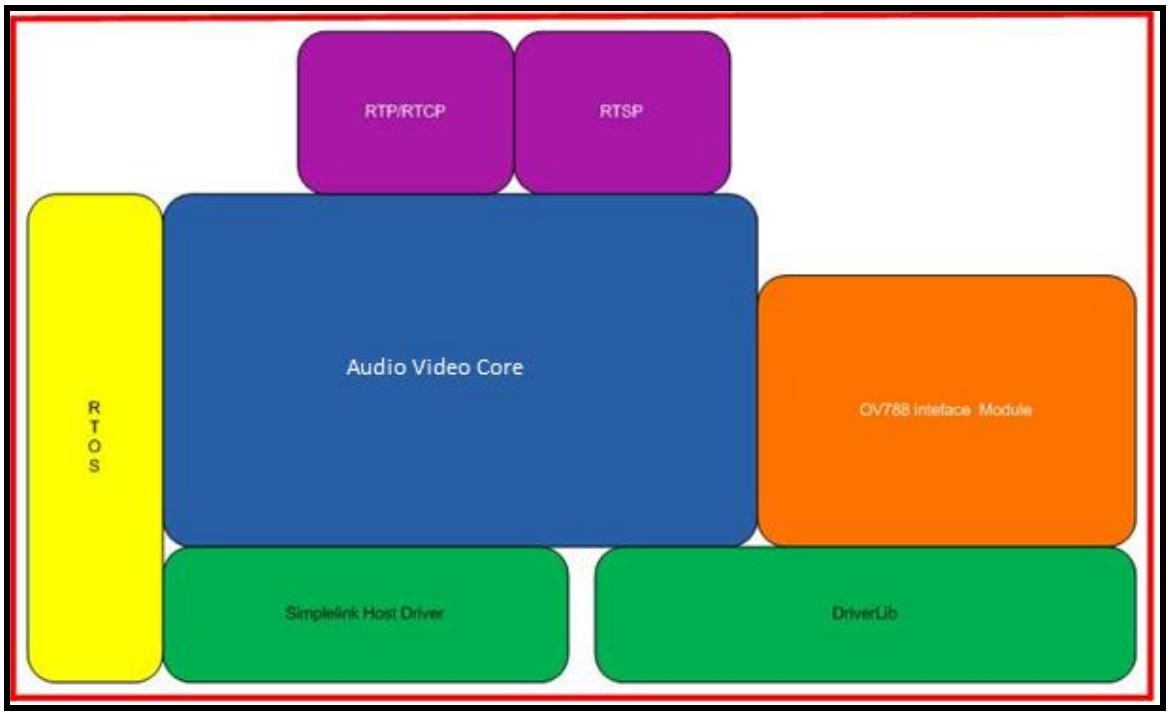
- SPI - 4 wire SPI (CS, CLK, MISO, MOSI) is used to for data exchange between CC3200 and OV788
- OVT_SYNC (GPIOA) - This signal is controlled by CC3200 and used to notify OV788 when CC3200 is ready to send/receive the data. Refer to 'SIF interface timing Diagram' for details.
- OVT_RDY (GPIOB) - This Signal is controlled by OV788 and used to notify CC3200 when the OV788 is ready to receive/send the data. Refer to 'SIF interface timing Diagram' for details.
- WLAN_ON - This signal is used to enable the 1A LDO. This signal should be asserted before enabling the NWP or powering on the OV788 subsystem.
- POWER_EN - This signal is used to power on the OV788 subsystem.

Copyright © 2016, Texas Instruments Incorporated

Embedded Software System

In the following subsections, the software architecture is described. The first subsection describes the setup of various software modules. Subsequently, modules are described in details. The entire software can be separated into multiple parts: Core application, RTSP library, RTCP/RTP Library, OV788 interface module.

The Software block diagram of CC3200_Video_Camera is shown below:



The Core application is responsible for processing the incoming request, managing the connection, querying the audio/video data from OV788 and sending it over to the remote client using RTSP/RTP/RTCP protocol.

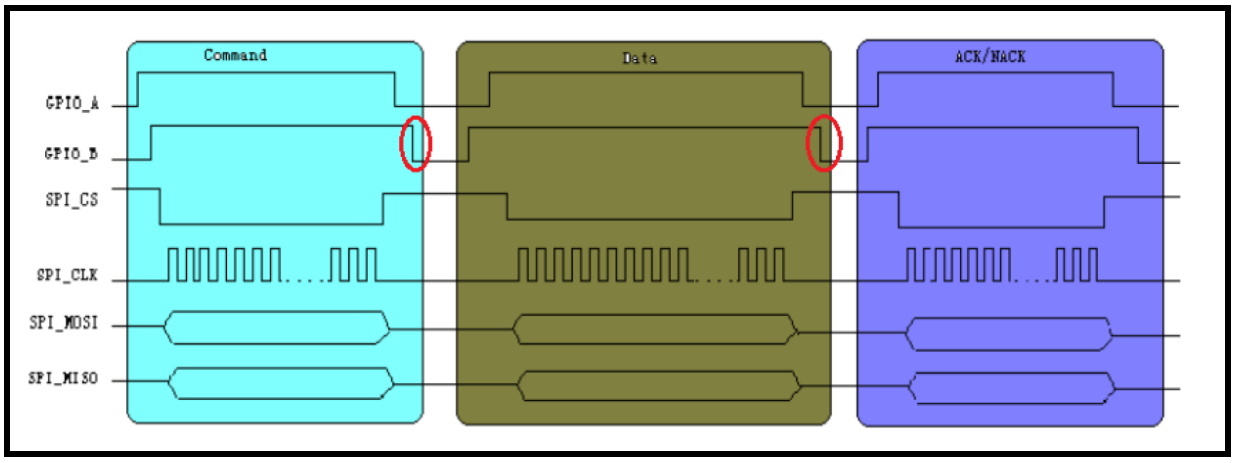
Software Components

OV788 Interface Library

OV788 library contains the implementation for communication with the OV788 module. Functions are provided to boot up module, download the OV788 firmware, configure the sensor and get the video and audio streams from OV788.

Copyright © 2016, Texas Instruments Incorporated

The device uses the SPI interface (master mode) along with two GPIO, SYNC (A) and RDY (B), to communicate with the OV788.



OV788 library that's provided w/ the package supports the following feature.

- Boot Up and initialize OV788 module
- Power off OV788 module
- Configure
- Frame Rate
- Resolution
- Brightness
- Contrast
- Saturation
- Frequency
- Flip
- Mode
- Enable/Disable Video streaming
- Get Video stream info
- Get Video stream data

Copyright © 2016, Texas Instruments Incorporated

- Get Audio stream info
- Get Audio stream data

For Library API details refer to ‘ov_sif_inerface’ API document @<cc3200-sdk>/docs

RTSP Library

Real Time Streaming Protocol (RTSP) is used for establishing and controlling media session between two systems. RTSP is generally used along with RTP/RTCP for media stream delivery.

RTSP library provided with the package contains the implementation of RTSP server, used for processing the RTSP client request and generates the responses to be sent to the client. This library supports the following protocol directives requests.

- OPTIONS: An ‘OPTIONS’ from client to server is used to request for the list of commands the server shall support/accept.
- DESCRIBE: A ‘DESCRIBE’ request contains the type of reply data that can be handled. The server reply contains the description of the media streams supported/controlled with the aggregated URL.
- SETUP: A ‘SETUP’ request specifies how a single media stream must be transported along with the local ports for the receiving RTP and RTCP data. This must be done before a PLAY request is sent by the client. The server’s reply confirms the chosen parameters and adds the server's chosen ports.
- PLAY: A ‘PLAY’ request is used to start the streaming of the media.
- TEARDOWN: A ‘TEARDOWN’ request is used to terminate the session. It stops all media streams and frees all session related data on the server.

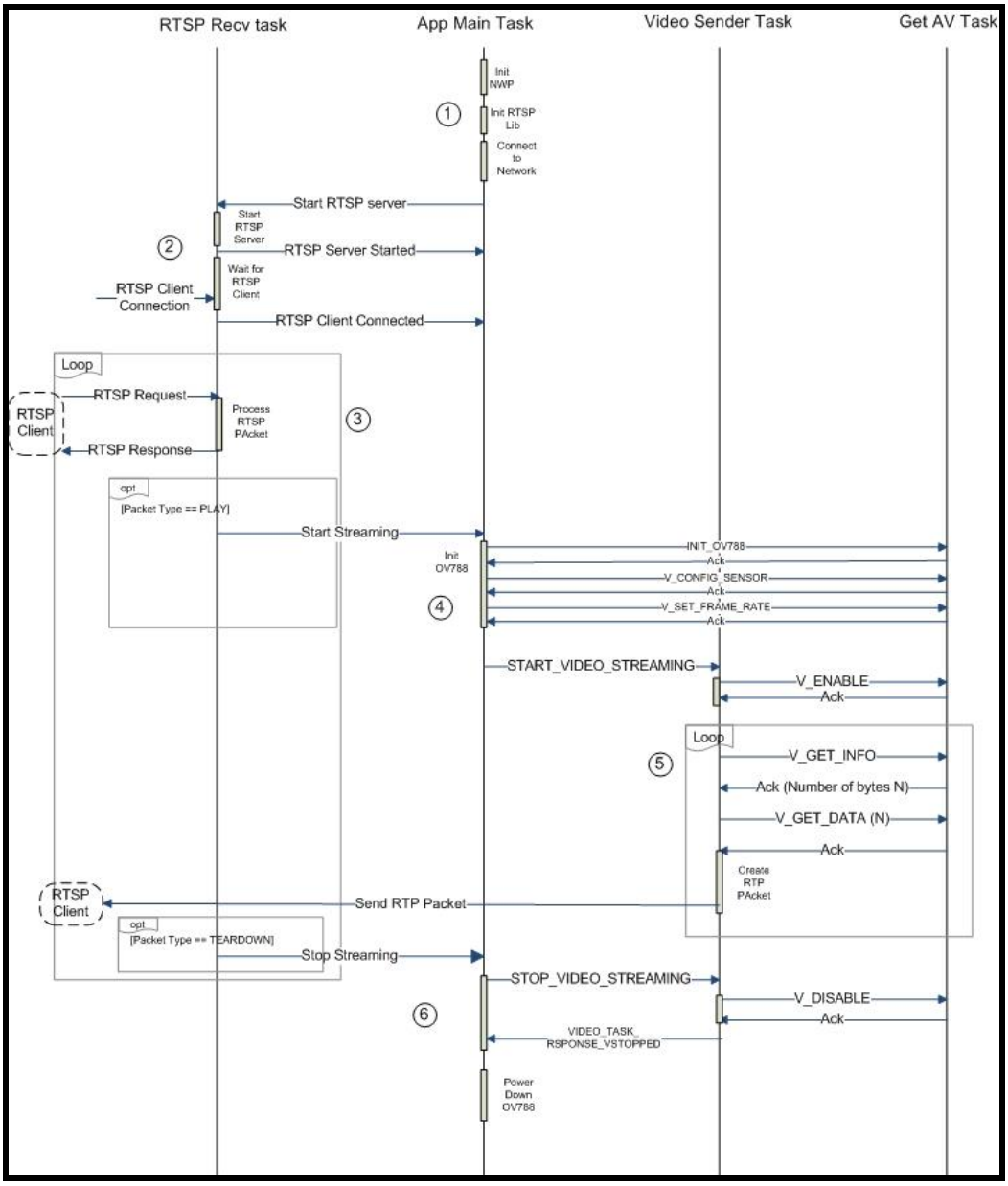
RTP/RTCP Library

Real-Time Transport Protocol (RTP) is used for delivering the media over the network. RTP is used in conjunction with the RTP Control Protocol (RTCP). RTCP is used to monitor the transmission statistics and quality.

Provided RTP library supports the profile for H.264 Video, RFC6184 and is used to create the RTP and RTCP packets for the mentioned profile. It also supports PCM L16 Audio

Copyright © 2016, Texas Instruments Incorporated

Application Flow



- 'App Main task' initializes the network processor, Initialize the RTSP library and connect the device to a network.
- Once the device is connected to network the 'RTSP Recv task' will start a TCP server and wait for RTSP client to connect.
- Once the client is connected and RTSP packet is received 'RTSP Rcv Task' will process the packet. If 'PLAY' packet is received a message is sent to 'App main task' to start the

Copyright © 2016, Texas Instruments Incorporated

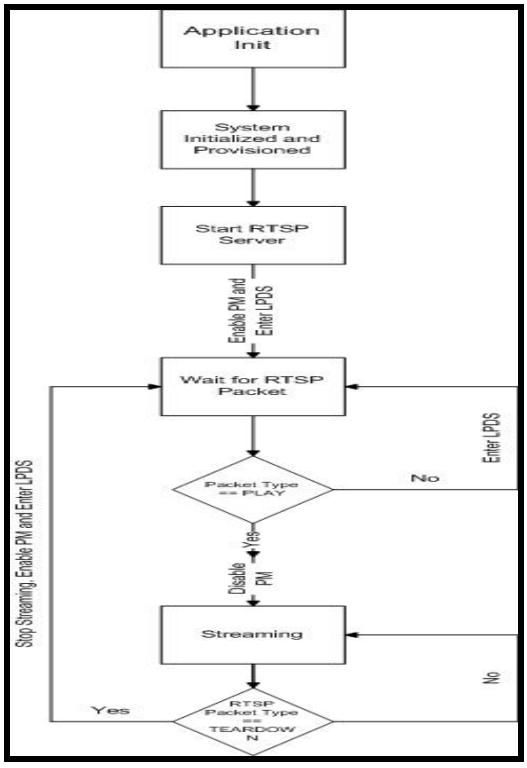
streaming. If ‘TEARDOWN’ packet is received a message is sent to ‘App main task’ to stop the streaming.

- Once start streaming message is received ‘App main task’ will power on the OV788 subsystem and download the OV788 firmware and configure the sensor. After initialization a message will be sent to ‘Video Sender Task’ and ‘Audio Sender Task’ to start the video and audio streaming respectively.

- Once Start Video streaming message is received ‘Video sender Task’ will send the ‘V_ENABLE’ message to enable the video on OV788. After enabling, the ‘Video sender task’ will request the available data info and on valid data info will get the video data in a loop until ‘Stop Video streaming’ message is received. The same sequence is followed for audio streams as well.

- If the Stop Streaming message is received by ‘App main task’, a message is sent to ‘Video Sender Task’ and ‘Audio Sender Task’ to stop streaming and wait for acknowledgement message. Once acknowledgement is received it will power down the OV788 subsystem.

Low Power Mode*



Copyright © 2016, Texas Instruments Incorporated

For low current consumption while the system is not streaming CC3200 MCU is configured in LDPS mode while the NWP is in Idle connected mode. This power management scheme is selected to allow the remote app to connect at any time and request for the video stream. Depending upon the use case other lower power mode can be used. For example, if an intermediate server is used for storing and streaming the data where the CC3200 application will be in client mode then the CC3200 can be put in HIB mode when idle and can be woken up using an interrupt to start the streaming.

Using 3AA Duracell battery system can stream video continuously for approximately 4 hours. For CC3200 current numbers please refer to the CC3200 Datasheet.

* - Support not enabled in this version of the application

Limitations

- There is a latency of ~2s in the streaming, mostly due to the buffering by the Smartphone application.
- VLC Application on Android doesn't process L16 payload, and therefore cannot be used for audio-video streaming – 'RTSP Player' can be used instead (on Android Smartphones) for simultaneous streaming of audio and video.
- VLC Application on iOS device (sometimes) fails to play the audio streams beyond ~15m, although the embedded application keeps streaming the data samples – This problem is not observed w/ 'RTSP Player' on iOS device.
- Sender reports when sent to applications like VLC were resulting in total audio silence/blackout – Therefore, processing of SR/RR RTCP packets is currently not enabled. This might result in out-of-sync AV when the application is run over prolonged period of time. These RTCP packet processing can be enabled by defining the macro 'RX_RR_TX_SR'
- The OV788 system is not power optimized while in idle mode.

References

- OV788 SIF Protocol Document

Copyright © 2016, Texas Instruments Incorporated

Appendix

RTSP Library APIs

API: ***rtsp_init***

This function is used to initialize the RTSP Library. This function should be called before any other RTSP library function		
Arguments	`p_param`	Pointer to buffer containing the source description information
Return		0 on success, negative value on error

API: ***rtsp_deinit***

Deinintilize the RTSP library		
Arguments	`None`	NA
Return		0 on success, negative value on error

API: ***rtsp_release_setup***

Resets the setup-counter that was incremented on receiving an SETUP packet		
Arguments	`None`	NA
Return		0 on success, negative value on error

API: ***rtsp_packet_parser***

This function processes the received RTSP packet and create the appropriate response packet		
Arguments	`p_in_buffer`	[Input] : Pointer to buffer containing RTSP packet
	`rtp_pkt_size`	[Input] : Size of the RTSP packet

Copyright © 2016, Texas Instruments Incorporated

	p_rtsp_session_config	[Input] : Pointer to structure (rtspModuleConfig) containing the information required to create RTSP response packet
	p_out_buffer	[Output] : Pointer to buffer for storing RTSP response packet
	rtsp_resp_pkt_len	[Input] : Output buffer size
	pkt_info	[Output] :Pointer to structure (rtspPacketInfo) containing the RTSP packet info
Return		Size of data in buffer, negative value on error

Structure: ***rtsp_session_config_s***

This structure is used to pass the information required to create the RTSP response packet		
stream_port[MEDIA_STREAMS]	rtsp_stream_port_s	Socket port pair for a media stream
session_timeout	Long	Server session timeout value
session_timeout	Long	Server session number
timestamp	Unsigned long	Current timestamp value
multicast_ip	Unsigned char array	Server Multicast Destination IP
date	Unsigned char array	Date and Time string

Structure: ***rtsp_pkt_info_s***

This structure is used to return the RTSP packet related information		
pkt_type	rtsp_packet_type_e	RTSP packet type
pkt_response	rtsp_pkt_response_u	Packet payload information
seq_num	Long	RTSP request sequence number
setup_count	Long	Total Setup requests

Copyright © 2016, Texas Instruments Incorporated

Structure: ***rtsp_setup_data_s***

This structure is used to return SETUP RTSP packet payload information		
`rtp_port`	Unsinged short	Client RTP port
`rtcp_port`	Unsinged short	Client RTCP port
`is_multicast`	char	Is the request for UDP multicast

RTP/RTCP Library APIs

API: ***rtp_init***

Inintilizes the RTP library		
Arguments	None	NA
Return	Long	Size of data in output buffer, negative value on error

API: ***rtp_deinit***

Deinintilizes the RTP library		
Arguments	None	NA
Return	Long	Size of data in output buffer, negative value on error

API: ***rtp_process_rtcp_pkt***

Process the RTCP packet and return the result		
Arguments	`p_input`	[Input] : Pointer to buffer containing RTCP packet
	`length`	[Input] : Size of the RTCP packet
	`p_out_data`	[Output] : Pointer to structure for storing the RR packet
Return		data size in output buffer, negative error-code on error

Copyright © 2016, Texas Instruments Incorporated

API: ***rtp_encode***

Create the RTP for video p_rtp_profile		
Arguments	p_data	[Input] : Pointer to data buffer
	data_len	[Input] : Size of the data in buffer
	p_out_buffer	[Output] : Pointer to buffer for storing the output RTP packet
	out_buffer_len	[Input] : Maximum size of output buffer
	p_rtp_profile	[Input] : Pointer to structure (rtpProfile) containing the required profile and RTP header specific information to create the RTP packet
Return		Size of data in output buffer, negative value on error

API: ***rtp_create_sr_pkt***

Create the sender report packet		
Arguments	p_output	Pointer to buffer containing the output SR
	buffer_len	[Input] : Maximum size of the output buffer
	sr_info	[Input] : Pointer to structure (rtcpSRinfo) containing the information required to create SR packet
Return		Size of data in output buffer, negative error-code on error

Structure: ***rtp_profile_s***

This structure is used to pass the RTP header and profile specific information		
p_data	rtp_profile_data_s	RTP Profile Data
type	profile_type_e	Profile type, Currently only video is supported
payload_format	payload_format_e	Payload format

Copyright © 2016, Texas Instruments Incorporated

timestamp	Unsigned long	Time stamp value
ssrc	Unsigned long	32 bit synchronization source identifier.
seq_num	Unsigned short	RTP packet sequence number
payload_type	Unsigned char	Data payload type, This is the value used in RTSP describe response.

Copyright © 2016, Texas Instruments Incorporated