



Multimedia framework user manual

The document describe how to use the multimedia framework



Table of Contents

| 1 | Intr | oduction Multi Media Framework (MMF)3 |
|----|---------|---|
| 2 | Imp | lement source and sink4 |
| | 2.1 | Media Source and Sink Module4 |
| | 2.2 | MMF API Documentation5 |
| | 2.3 | MP3 API Documentation6 |
| | 2.4 | Mp3 Config Table Detail and supported frequencies7 |
| | 2.4 | .1 Key Properties of an Mp3 File7 |
| | 2.4 | .2 Mp3 Frequency table7 |
| Но | ow to I | Run these Examples8 |
| | 2.5 | Media Single Stream – Source UVC Camera, Sink RTSP Stream8 |
| | 2.6 | Media Single Stream – Source I2S, Sink RTSP Stream15 |
| | 2.7 | Media Single Stream – Source UVC Camera, Sink SD Card16 |
| | 2.8 | Media Single Stream – Source RTP Stream, Sink I2S18 |
| | 2.9 | Media Multiple Streams – Source GEO Camera, Sink RTP Stream |
| | 2.10 | Media Multiple Streams – Source GEO Camera, Sink SD Card |
| | 2.11 | MP3 Media Streaming Over RTP (Source-RTP; Sink Mp3 I2S)23 |
| | 2.12 | MP3 Play Audio from SDCARD27 |



1 Introduction Multi Media Framework (MMF)

The Multimedia Framework (MMF) is used to handle all sorts of media applications on Ameba. The MMF structure is as shown in the following chart.



There are two important entities in the MMF. One is the source that produces the resource and the other is sink that consumes the resource. The source can be the file input, UVC camera, or storage; the sink can be RTSP or other stream.

In order to use the MMF, the following aspects must be fulfilled.

- Define a valid source
- Define a valid sink
- Create OS message queues in application level for both source and sink module to access. Module-specific configuration may be made if required.
- Call the MMF APIs to start the multimedia streaming.





2 Implement source and sink

MMF allows users to define our own source and sink modules depending upon the application. Although implementation details may be different, these basic rules of the MMF structure should be a little bit similar. This section shall explain how to create a source and a sink in MMF using the example UVC source and RTSP sink. Further explanations on how to run the other examples are discussed in Section 3.

2.1 Media Source and Sink Module

MMF requires users to predefine both source and sink modules that implement create, destroy, set_param, and handle function callbacks.

| Media Source Module | Media Sink Module |
|--------------------------------------|-------------------------------------|
| typedef struct _media_source_module{ | typedef struct _media_sink _module{ |
| void* (*create)(void); | void* (*create)(void); |
| void (*destroy)(void*); | void (*destroy)(void*); |
| int (*set_param)(void*, int, int); | int (*set_param)(void*, int, int); |
| int (*handle)(void*, void*); | int (*handle)(void*, void*); |
| }msrc_module_t; | }msink_module_t; |

- (*create): pointer to the function that loads and initializes the source/sink stream of the module. For example, for UVC source, it points to the function in which the UVC driver is initialized and the corresponding context is returned.
- (*destroy): pointer to the function that de-initializes module instance and releases resource.
 For example, for UVC source, it points to function in which UVC driver is de-initialized and the corresponding context is released.
- (*set_param): pointer to function that sends the control command to the MMF layer or a specific module. For example, for UVC source, it points to function that controls UVC parameters ("frame height", "frame width", "framerate", etc.) and MMF service task on/off.
- (*handle): pointer to the function that manipulates media data (how to produce data in source or how to consume data in sink). Data is transferred from source to sink and vice versa by means of OS message queue. Please note that MMF service task reacts differently based on message exchange buffer status. The status codes are introduced as follows:

| 0 | #define STAT_INIT | 0 | //indicating buffer using at first time |
|---|-----------------------|---|---|
| 0 | #define STAT_USED | 1 | //indicating buffer consumed by sink |
| | | | //(must update this status in sink handle) |
| 0 | #define STAT_READY | 2 | //indicating buffer ready for use by source |
| | | | //(must update this status in source handle) |
| 0 | #define STAT_RESERVED | 3 | //indicating buffer being processed or |
| | | | <pre>//reserved and can't be accessed right now</pre> |

After msrc_module_t and msink_module_t structure are properly defined, user should declare the customized source and sink modules in mmf_source_list.h and mmf_sink_list.h, respectively. Thus, MMF can be aware of new-defined modules and these customized modules



can be executed. Some built-in source and sink module definitions can be found under the path 'component\common\media\framework'.

2.2 MMF API Documentation

This section discusses the usage of MMF APIs.

msrc_context* mmf_source_open(msrc_module_t* source)

Create MMF source context instance.

msrc_module_t* source: specify which source module class to be created. Return value: a pointer to source context instance.

void mmf_source_close(msrc_context* ctx)

Release the resource related to source context instance and lower level memory. msrc context* ctx: pointer to source context instance that is ready to be released.

int mmf_source_ctrl(msrc_context* ctx, int cmd, int arg)

Set control command to a specific source context instance.

msrc_context* ctx: pointer to a specific source context instance.

int cmd: the input command code, needs to be defined in mmf_common.h.

int arg: relevant value along with the input command code. Pass 0 if not applicable.

int mmf_source_get_frame(msrc_context* ctx, exch_buf_t *exbuf)

Prepare one frame or block of data and add its reference to message exchange buffer. msrc context* ctx: pointer to a specific source context instance.

exch_buf_t *exbuf: pointer to the message exchange buffer used to convey media frame.

msink_context* mmf_sink_open(msink_module_t *sink)

The functionality is similar to "mmf_source_open".

void mmf_sink_close(msink_context* ctx)

The functionality is similar to "mmf_source_close".

int mmf_sink_ctrl(msink_context* ctx, int cmd, int arg)

The functionality is similar to "mmf_source_ctrl".

int mmf_sink_put_frame(msink_context* ctx, exch_buf_t* exbuf)

Dereference one frame or block of data from message exchange buffer. msink_context* ctx: pointer to a specific sink context instance.



exch_buf_t *exbuf: pointer to the message exchange buffer used to convey media frame.

2.3 MP3 API Documentation

This section discusses the usage of MP3 APIs.

mp3_decoder_t mp3_create(void);

Creates the mp3 decoder instance and allocates required memory.

Input Parameter: No input parameters required

Return value: a pointer to the mp3_decoder_t structure. This must be created locally in the application before calling the create function.

int mp3_decode(mp3_decoder_t *dec, void *buf, int bytes, signed short *out, mp3_info_t *info);

After creation of the mp3 decoder instance use this API to initiate the actual decoding process. Input Parameters:

 \rightarrow mp3_decoder_t* dec:-Pass the mp3 decoder instance that was obtained using the create function mentioned before.

 \rightarrow void* buf:- The buffer pointer to the input frame of the mp3 data that needs to be decoded. \rightarrow int bytes:- The number of bytes in the input mp3 data buffer.

→ signed short *out:- Pointer to the buffer where the decoded mp3 data will be present after the decoding process is over.

→mp3_info_t *info:- This structure is used to identify the frequency of the decoded mp3 data and number of channels and the number of decoded bytes in each frame of the output data, details regarding this structure is found in the g711_decoder.h file.

Return Value:

Int, this integer returns the actual number of bytes that were successfully decoded from the input buffer provided to the mp3 decode function, based on this value the part of the input buffer that was not decoded must be provided again in the next iteration for decoding.

void mp3_done(mp3_decoder_t *dec);

Release the memory allocated for the mp3 instance and delete the mp3 instance.

Input Parameters:

mp3_decoder_t* dec :- The original instance created using mp3_create needs to be passed in as the input to this parameter.

Output Parameter:

None



2.4 Mp3 Config Table Detail and supported frequencies

In order to use the Mp3 decoder and hear the output audio in a smooth and uninterrupted manner the I2S driver must be configured with the right output frequency, the number of channels in the audio data and the I2S DMA page size must match the exact number of decoded bytes per frame of the mp3 in order to hear clear and uninterrupted audio from the I2S output.

2.4.1 Key Properties of an Mp3 File

• <u>Bit-Rate</u>

This determines the number of bits contained in one second of the mp3 file. The bit rate of the mp3 file determines the size of each input frame. The input frame is decoded by the mp3 decoder to produce the output buffer that contains the raw audio data.

<u>Frequency</u>

This determines the sampling frequency of the input audio signal. This is an important parameter that determines the size of the decoded output frame. In case the audio is being played by means of I2S, the I2S driver must be initialized to the right frequency in order to be able to hear the audio clearly.

<u>Number of Channels</u>

An mp3 audio file can have one or two channels; this parameter determines the audio that needs to be played in the left and right channel respectively.

The mp3 audio data properties can be checked using any media player. Most common media players like vlc and Audacity can be used to check the frequency and number of channels present in the mp3 file.

2.4.2 Mp3 Frequency table

This frequency table helps map the frequency and the number of channels in the mp3 files to the size of the output buffer that needs to be used to accumulate the decoded output of the mp3 decoder. The frequencies supported by the mp3 decoder are listed below.

| Frequency(hz) | Channels | Decoded Bytes |
|---------------|----------|---------------|
| 8000 | 1 | 1152 |
| 8000 | 2 | 2304 |
| 16000 | 1 | 1152 |
| 16000 | 2 | 2304 |
| 22050 | 1 | 1152 |
| 22050 | 2 | 2304 |
| 24000 | 1 | 1152 |



| 24000 | 2 | 2304 |
|-------|---|------|
| 32000 | 1 | 2304 |
| 32000 | 2 | 4608 |
| 44100 | 1 | 2304 |
| 44100 | 2 | 4608 |
| 48000 | 1 | 2304 |
| 48000 | 2 | 4608 |

In order to hear clear audio from the examples, the I2S driver should be initialized to the right frequency and the I2S DMA page size must be set to the value of decoded bytes by using the table shown above. These details are provided in the start of the mp3 example as well as shown below.

| /*+ Frequency(hz) | + Number of Chann (CH_MONO:1 CH_SI | els Decode EREO:2) (125_D) | ed Bytes MA_PAGE_SIZE) | | |
|------------------------|--|----------------------------------|-----------------------------|-------------------|----------------------------|
| + | / I | 1 | 1152 | | |
| 8000 | | 2 1 | 2304 | | |
| 16000 | | 1 1 | 1152 | | |
| 16000 | | 2 1 | 2304 | | |
| 22050 | | 1 | 1152 | | |
| 22050 | | 2 | 2304 | | |
| 24000 | | 1 1 | 1152 | | |
| 24000 | | 2 | 2304 | | |
| 32000 | | 1 | 2304 | | |
| 32000 | | 2 1 | 4608 | | |
| 44100 | | 11 | 2304 | | |
| 44100 | | 2 | 4608 | | |
| 48000 | 1 | 1 | 2304 | | |
| 48000 | 1 | 2 | 4608 | | |
| + | + | + | +*/ | | |
| // | | CONFIG | Parameters | | / |
| #define 125_DMA_1 | PAGE_SIZE 4608 | //Use frequency m | mapping table and | set this value | to number of decoded bytes |
| | | //options:- 1152, | , 2304, 4608 | | |
| fderine NUM_CHAN | NELS CH_STEREO | //USe mp3 file pi | roperties to dete | rmine number or . | cnanneis |
| | | //Options:= Ch_M | JNO, CH_SIEREO | | |
| Fderine SAMPLING | TREQ SR_32KH2 | //use mp3 rile pi | roperties to iden | city frequency a. | na use appropriate macro |
| | | //options:- SR_BI | KHZ =>8000hz | - PASS | |
| | | // SR_10 | 5KHZ =>16000hz | - PASS | |
| | | // SR_24 | 2VUZ ->224000hz | - 2400 | |
| | | // SR_32 | 2KR2 ->32000NZ | - FADD | |
| | | // SR_48 | ->48000nz | - 1400 | |
| | | // CD 0/ | C10770 0.C000b | NOT CURRONTER | |

How to Run these Examples

The following sections provide the examples of using the MMF and can be used as references to create customized applications.

2.5 Media Single Stream – Source UVC Camera, Sink RTSP Stream

Please refer to following steps to understand how to run the MMF example with source as UVC camera and sink as RTSP stream.

Pre-requisites

- Ameba board
- UVC camera (E.g., Logitech C170)
- USB cable is used to connect with PC.



- WiFi connection for sending RTSP streaming out
- USB OTG cable is used to connect UVC camera to Ameba board.

Hardware Setup

• Connect the mini-USB OTG cable to the port, labelled as "con3", on the underside of the Ameba board as shown in the following figure.



• Next, connect the USB camera to the mini-USB OTG cable which is connected with the Ameba board as shown in the following figure.



• Finally, connect the standard micro-USB cable to the port, labelled as "con1", on the upper side of the Ameba board, as shown in the figure bellow, and connect the other end to the PC.





Software Setup

• Once the hardware setup is completed as mentioned above, please refer to the software setup steps in order to run the example. First, please open the IAR workspace, Project.eww in EWARM-RELEASE folder, and then the default workspace, Debug, is loaded as shown in the following figure.

| /orkspace | | × |
|---------------------|------|-----|
| Debug | | - |
| Files | 4- | BC. |
| | C.m. | 10 |
| application - Debug | ~ | - |
| | | |
| | | |
| | | |
| | | |
| | | |
| | ~ | |
| | • | |
| | | |
| L 🕀 🗋 Output | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | - |
| application | | |

• In order to use the UVC features, please select the UVC workspace. In the list box, please select the UVC option as shown below.





| <u>File Edit View Project CMSIS-DAP Tools Window H</u> | lelp |
|--|------|
| · · · · · · · · · · · · · · · · · · · | |
| Workspace × | |
| Debug 🗸 | |
| Debug | |
| INIC_SPDIO | |
| | |
| Console | |
| | |
| | |
| HT Concerning to the second se | |
| H⊞ DSDRAM ✓ | |
| H 🕀 🗀 user | |
| 🗕 🖽 🗀 utilities | |
| Left Output | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| application | |

After the UVC workspace is selected, the appropriate example, part of the workspace, needs to be enabled. Please open the platform_opts.h file in IAR IDE as shown below. Then, please set the "CONFIG_EXAMPLE_MEDIA_SS" to be "1"as shown in following figure and, thus, the media single stream example is enabled. The example code can be found in the project path, utilities→example→example_media_ss.c.

| Elle Edit View Project CMSIS-DAP Iools Window Help | | | | | | |
|--|---------|----------------|--|------------|--|--|
|) 🗅 🚅 🖬 🎒 🎒 👗 🖻 🛍 🗠 🗠 | | | - 🗸 🦻 🐂 🛺 🗗 🖻 🐢 📣 🚱 🔤 🥨 👷 🥭 🕭 | | | |
| Workspace | × | platform | n_opts | | | |
| UVC | - | # | define FATFS DISK SD 1 | | | |
| Files | 50 BF v | | endif | П | | |
| He peripheral | • | | endif | JL | | |
| HE SDRAM | × • | I [• | endli | V | | |
| | | | | ¥ 1 | | |
| | • • | - I ÷ | define CONFIG_EXAMPLE_MEDIA_SS | 1 | | |
| | | | deline config_example_media_ms | 0 | | |
| autocont.n | | | define CONFIG_EXAMPLE_MEDIA_GEO_RTP | 0 | | |
| basic_types.n | | | define CONFIG EXAMPLE MEDIA GEO MP4 | 0 | | |
| Diag.n | | | / Use media source/sink example | | | |
| DLib_Config_Normai.n | | I ⊨ ∔ | if (CONFIG EXAMPLE MEDIA SS==1) (CONFIG EXAMPLE MEDIA MS= | =1) | | |
| DLib_Defaults.n | | | undef CONFIG INCLUDE SIMPLE CONFIG | -/ // | | |
| DLID_Product.n | | | define CONFIG INCLUDE SIMPLE CONFIG | | | |
| DLib_Product_stillg.in | E | | define CONFIG_INCLODE_SIMPLE_CONFIG 0 | | | |
| Sevemple entry h | | | GETTINE CONFIG_ENABLE_WPS 0 | | | |
| EreeBTOSh | | 부 🗄 | IT CONFIG_EXAMPLE_MEDIA_GEO_MP4 | | | |
| FreeBTOSConfig h | | | define FATFS_DISK_SD 1 | | | |
| hal misch | | † † | endif | | | |
| intrinsics h | | # | endif | | | |
| | | | | | | |
| main h | | | * For Mjpeg capture example*/ | | | |
| mpu wrappers h | | | define CONFIG EXAMPLE MJPEG CAPTURE 0 | | | |
| D plotform outcoonf h | | ⊨ ∔ | if CONFIG EXAMPLE MJPEG CAPTURE | | | |
| - Delatform opts.h | | | define FATES DISK SD 1 | | | |
| A poncole in | | | and f | | | |
| portmacro.h | | | OUALT | | | |
| projdefs.h | | | | | | |
| | | | For EAP method example ************************************ | */ | | |
| Stdint.h | - | | define CONFIG_EXAMPLE_EAP 0 | | | |
| application | | | | | | |
| 1 | | 1 () | III. | | | |

 Once all the configurations have been satisfied, please right click on project name, applications - UVC, in the left sidebar of IAR IDE as shown below and select "Rebuild All" option to build this project.



| <u>File Edit View Project CMSIS-DAP Tools Window E</u> | <u>d</u> elp | | | | | |
|--|--|--|--|--|--|--|
| ● 🖬 🕼 🖉 🕼 🛍 🗠 ⇔ 🔹 🔹 🔹 🔹 🚽 🔸 👘 🖉 🖉 🖉 🖉 🖉 🖉 🖉 🖉 | | | | | | |
| Workspace × | platform_opts | | | | | |
| UVC - | #define FATFS DISK SD 1 | | | | | |
| Files 😤 📴 | #endif | | | | | |
| 🗆 🗇 applicatio - 💷 🕐 | - #endif | | | | | |
| Options | - #endif | | | | | |
| Make | | | | | | |
| | thefine CONFIG EXAMPLE MEDIA SS | | | | | |
| B OS Rebuild All | #define CONFIG_EXAMPLE_MEDIA_MS 0 | | | | | |
| - Deriphera Clean | tdefine CONFIG_EXAMPLE_MEDIA_GEO_RTP 0 | | | | | |
| | // Use media source/sink example | | | | | |
| | = #if (CONFIG EXAMPLE MEDIA SS==1) (CONFIG EXAMPLE MEDIA MS==1) | | | | | |
| Here Contract Add → | #undef CONFIG INCLUDE SIMPLE CONFIG | | | | | |
| | #define CONFIG_INCLUDE_SIMPLE_CONFIG 0 | | | | | |
| Remove | #define CONFIG_ENABLE_WPS 0 | | | | | |
| Rename | <pre>#if CONFIG_EXAMPLE_MEDIA_GEO_MP4</pre> | | | | | |
| Version Control System | <pre>#define FATFS_DISK_SD 1 #endif</pre> | | | | | |
| Open Containing Folder | - #endif | | | | | |
| File Properties | | | | | | |
| Colora Antina | /* For Mjpeg capture example*/ | | | | | |
| Set as Active | #define CONFIG_EXAMPLE_MJPEG_CAPTURE 0 | | | | | |
| | = #if CONFIG_EXAMPLE_MJPEG_CAPTURE | | | | | |
| | #define FATFS_DISK_SD_1 | | | | | |
| | #eudit | | | | | |
| | /************************************* | | | | | |
| | <pre>#define CONFIG_EXAMPLE_EAP 0</pre> | | | | | |
| application | ↓ | | | | | |

Flashing and Testing

 Before flashing the Ameba board, please set up a serial monitor via Tera Term or PuTTY and set up serial port to be COMX/38400, X: Port number. Once the build process is done and the Ameba board is connected, the flashing can be started by clicking the following in IAR IDE.

 $Project \rightarrow Download \rightarrow Download Active Application$

File Edit View Droject CMSIS-DAD Tools Window H

| The Part Tran | (Toldard) Warner and Theorem Theorem | |
|----------------|--------------------------------------|--|
| | Add Files | ▼ イ ト た 江 図 D P P P @ B B 開 開 解 S D D |
| Workspace | Add Group | 8 |
| UVC | Import File List | fine FATFS DISK SD 1 |
| Files | Add Project Connection | pdif |
| 🗆 🗐 applicatio | Edit Configurations | ldif |
| - E Consis | Remove | ldif |
| H-B 🖸 lib | Create New Project | fine CONFIG_EXAMPLE_MEDIA_SS 1 |
| | Add Existing Project | fine CONFIG_EXAMPLE_MEDIA_MS 0 |
| - Deriphe | Outines Alt. 17 | Fine CONFIG EXAMPLE MEDIA GEO MDA |
| - SDRAM | Options Ait+F/ | Use media source/sink example |
| User | Version Control System | (CONFIG EXAMPLE MEDIA SS==1) (CONFIG EXAMPLE MEDIA MS==1) |
| | Make F7 | def CONFIG INCLUDE SIMPLE CONFIG |
| | Compile Ctrl+F7 | fine CONFIG_INCLUDE_SIMPLE_CONFIG 0 |
| | Rebuild All | fine CONFIG_ENABLE_WPS 0 |
| | Clean | CONFIG_EXAMPLE_MEDIA_GEO_MP4 |
| | Batch build | Fine FATFS_DISK_SD 1 |
| | | dif |
| | Stop Build Ctrl+Break | |
| | Download and Debug Ctrl+D | For Mjpeg capture example*/ |
| | Debug without Downloading | fine CONFIG_EXAMPLE_MJPEG_CAPTURE 0 |
| | Make & Restart Debugger Ctrl+R | CONFIG_EXAMPLE_MJPEG_CAPTURE |
| | Restart Debugger Ctrl+Shift+R | fine FATFS_DISK_SD 1 |
| | Download + | Download active application |
| | SFR Setup | Download file P method example ************************ |
| | Open Device Description File | crase memory U |
| application | Save List of Registers | W |



• The download and flashing process consumes more time. Please wait a moment. While the process is done, Please compare the messages in the Debug Log Window of the IAR project with that in the figure bellow. If the messages are similar or the same, the software is successfully flashed onto the Ameba board.

| application | <pre>/* For Mjpeg capture example*/ tdefine CONFIG_EXAMPLE_MOPEG_CAPTURE 0 #if CONFIG_EXAMPLE_MOPEG_CAPTURE tdefine PATFS_DISK_SD 1 fendif /************************************</pre> |
|--|--|
| Log Tue Sep 27, 2016 16:25:51: — Prepare hardw Tue Sep 27, 2016 16:25:53: 0x40000014=00000 Tue Sep 27, 2016 16:25:55: Loaded debugee: Tue Sep 27, 2016 16:25:56: Target Test | re for Flashloader |

- Uie Sep 27, 2016 18:25-56: largetreset
 Tue Sep 27, 2016 18:25-46: largetreset
 Tue Sep 27, 2016 18:27-46: Unodaded manor file: E\RELEASE Copy/project/vealtek_ameba1_va0_example/EWARM-RELEASE_\.\ component/soc/vealtek/8195a/misc/var_utility/common/flashloader/FlashRTLB195aMP mac
 Tue Sep 27, 2016 18:27-46: Loading the jety/TAGjet driver
 Tue Sep 27, 2016 18:27-44: Loading the jety/TAGjet driver
 Tue Sep 27, 2016 18:27-46: Loading the jety/TAGjet driver
- Next, please press the reset button on the Ameba board to restart the system. The log messages are shown as that in the following figure via Tera Term or PuTTY.



• Next, it is going to set Ameba board to connect with an AP for RTSP streaming purpose. Please use the following AT commands to join a WiFi network via Tera Term or Putty. The following figure is an example for your reference.

ATW0=<Name of WiFi SSID>: Set the WiFi AP to be connected ATW1=<Password>: Set the WiFi AP password ATWC: Initiate the connection



Streamon successful #ATWO=RealKungFu [ATW0]: AT_WLAN_SET_SSID_ [RealKungFu] [MEM] After do cmd, available heap 31472 # ATW1=RealTech [ATW1]: _AT_WLAN_SET_PASSPHRASE_ [RealTech] [MEM] After do cmd, available heap 31528 # ATRTW API: Join bss timeout [ATWC]: _AT_WLAN_JOIN_NET_ Joining BSS by SSID RealKungFu... RTL8195A[Driver]: set ssid [RealKungFu] RTL8195A[Driver]: start auth to 9c:1c:12:13:cd:00 RTL8195A[Driver]: auth success, start assoc RTL8195A[Driver]: association success(res=4) RTL8195A[Driver]: set pairwise key to hw: alg:4(WEP40-1 WEP104-5 TKIP-2 AES-4) RTL8195A[Driver]: set group key to hw: alg:4(WEP40-1 WEP104-5 TKIP-2 AES-4) keyid:1 wlan_wrtie_reconnect_data_to_flash():not the same ssid/passphrase/channel, write new profile t o flash Connected after 6343ms. Interface 0 IP address : 172.25.23.78 WIFI initialized init_thread(53), Available heap 0x7878 Interface 0 IP address : 172.25.23.78 Got IP after 7377ms. [MEM] After do cmd, available heap 36064 # connect successful sta mode rtsp stream enabled

 While the log message, "RTSP stream enabled", is shown in the log console, please open the VLC media player, click "Media→Open Network Stream" option in menu bar, and type in: "rtsp://xxx.xxx.xxx.xxx:yyy/test.sdp" before clicking play button. xxx.xxx.xxx: the Ameba IP address.

yyy: RTSP server port number (default is 554).

Also, please select 'Show more options' checkbox if advanced setting is required.



| Media Playback Audio Video Subtijle Topols View Help Image: Comparison of the comparison of th | A r | tsp://192.168.0.102:554/test.sdp - | VLC media player | IN THE OWNER OF THE OWNER | | 🛓 Open Media | | | |
|---|------|--|---|---------------------------|------|--|--|-------------------------------|-------------------|
| Image: Start Star | | dia Playback Audio Yideo Open Eile Open Eolder Open Devenk Stream Open Qapture Device Open Qapture Device Open Qacture Device Open Qactor Hedia Save Playlist to Eile Convert / Save Stream Quit at the end of playlist Quit | Subtile Tgols Ctrl+O Ctrl+Shift+O Ctrl+Shift+O Ctrl+F Ctrl+V Ctrl+C Ctrl+V • Ctrl+Y • Ctrl+S • Ctrl+S • | View Help | 7 | File Vetwork Prov Please enter (stgs://isse http://www. http://www. http://www. | Disc ** Network ** tocol a network URL: 600 U02554/seting resample.com/steam.asi 34 essample.com/steam.asi essample.com/steam.asi essample.com/steam.asi | Capture Device | |
| Bdit Options metvork-caching=1000 | 00:0 | | | T | 0000 | Show more o Caching Play anothe MRL Edit Options | ptions 1000 ms er media synchronously (extra aud rtsp.//192.168.0.102.554/set.a metwork-caching=1000 | Start Time io file,) Ip | 00H 00m 00e 000 🔄 |

2.6 Media Single Stream – Source I2S, Sink RTSP Stream

Please refer to following steps to understand how to run the MMF example with source as ALC5651 in I2S codec and sink as RTSP stream.

Pre-requisites

- Ameba board
- ALC5651 Ext. board
- Audio connection cable (3.5mm)
- USB cable is used to connect with PC.
- WiFi connection for sending RTSP streaming out
- VLC media player on the testing PC

Procedures

- Please set "CONFIG_EXAMPLE_MEDIA_SS" to be "1" in "platform_opts.h". The example code can be found in the project path, utilities → example → example_media_ss.c.
- Please set "SRC_I2S" to be "1" in "example_media_ss.c" and other source definitions are set to be "0" as shown in the following figure.

```
// select source
#define SRC_UVC 0
#define SRC_MJPG_FILE 0
#define SRC_H264_FILE 0
#define SRC_AAC_FILE 0
#define SRC_PCMU_FILE 0
#define SRC_I2S 1
```

In addition, the source code of SRC_I2S is in

mmf_source_i2s_file.[c/h](component\common\media\framework\mmf_source_modu les)



- Please compile the project and download the application into Ameba board as described in the previous section.
- Next, please connect the ALC5651 Ext. board with Ameba board. Then, please also connect the audio connection cable to line-in, labelled as "PH3", on ALC5651 Ext. board.
- As described in the previous section, please connect both the testing PC and Ameba to the same WiFi AP. As a result, the following figure can be shown. While the log message, "rtsp stream enabled", is shown in the log console, it means that the RTSP server is ready.

| Interface 0 IP address : 172.25.23.90 |
|--|
| Got IP after 3321ms. |
| [MEM] After do cmd, available heap 40080 |
| # connect successful sta mode |
| rtsp stream enabled |

 Next, please open the VLC media player, click "Media→Open Network Stream" option in menu bar, and type in: "rtsp://xxx.xxx.xxx" before clicking play button. xxx.xxx.xxx: the Ameba IP address.

Also, please select 'Show more options' checkbox if advanced setting is required.

| 🛓 Op | en Media File 💿 Disc 📲 Network 👹 Capture Di | evice |
|------|--|--------------|
| N | etwork Protocol | |
| P | ease enter a network URL: | |
| | http://www.example.com/stream.avi rtp://@i1234 mmsi/mms.examples.com/stream.asx rspi/serve.example.org/860(test.sdp http://www.yourube.com/watch?v=gg64x | |
| Sł | ow more options | Play Cancel |

2.7 Media Single Stream – Source UVC Camera, Sink SD Card

Please refer to following steps to understand how to run the MMF example with source as UVC camera and sink as SD card.

Pre-requisites

• Ameba board



- UVC camera
- USB cable is used to connect with PC.
- SD card and SD/MMC card connector

Procedures

- Please refer to the document (UM0073 Realtek Ameba-1 Storage User Manual.pdf) to follow the steps in Section 2 to connect SD/MMC card connector with Ameba board. Then, please also plug SD card into SD/MMC card connector.
- Please set "CONFIG_EXAMPLE_TIMELAPSE" to be "1" in "platform_opts.h". The example code can be found in the project path, utilities → example → example_media_tl.c.
- Please compile the project and download the application into Ameba board as described in the previous section.
- While the log message, "Streamon successful", is shown in the log console as the figure bellow, it is going to write jpeg files into SD card.



Please wait a while then pull out the SD card from the SD card connector on the Ameba board. Then, please also connect SD card with another PC or laptop to check whether the jpeg files exist or not, as shown in the figure bellow. Then, please download "ffmpeg.exe" from FFmpeg website and execute the batch file, "lapse.bat", in the path (component\common\example\media_time_lapse) to generate a time-lapse mp4 file.

| | ///////////////////////////////////// | | | | ▼ ◆ | |
|--------------|---------------------------------------|------------------|--------------|--------|-------|-----|
| 合管理 👻 📰 開啟 | 列印 新增資料夾 | | | | 800 - | . 0 |
| 我的最爱 | 名稱 | 修改日期 | 類型 | 大小 | | |
| Deskton | | 2016/2/2 Life 12 | IDEC BIO | 25 VD | | |
| Downloads | ings/.jpeg | 2010/2/2 _ + 12 | IDEC BUD | 35 KB | | |
| RecentPlacer | imgsa.jpeg | 2010/2/2+ 12 | IDEC BUG | 35 KB | | |
| Recentraces | imgss.jpeg | 2010/2/2 1 + 12 | IDEC BUG | 35 KB | | |
| W 10 (T | img40.jpeg | 2010/2/2 1 + 12 | IDEC BUR | 33 ND | | |
| | img41.jpeg | 2010/2/2 17 12 | IDEC BUR | 33 ND | | |
| | img42.jpeg | 2010/2/2 17 12 | IDEC BUR | 33 ND | | |
| り首張 | img45.jpeg | 2010/2/2 17 12 | IDEC BOR | 33 ND | | |
| 視訊 | img44.jpeg | 2016/2/2 上午 12 | JPEG 影像 | 35 KB | | |
| 國局 | img45.jpeg | 2016/2/2 上午 12 | JPEG 影像 | 35 KB | | |
| | img46.jpeg | 2016/2/2 上午 12 | JPEG 影像 | 35 KB | | |
| 電腦 | img47.jpeg | 2016/2/2 上午 12 | JPEG 影像 | 35 KB | | |
| WIN7 (C:) | img48.jpeg | 2016/2/2 上年 12 | JPEG 影像 | 35 KB | | |
| DATA (D:) | img49.jpeg | 2016/2/2 上平 12 | JPEG 影像 | 35 KB | | |
| MBED (E:) | img50.jpeg | 2016/2/2 上年 12 | JPEG 影像 | 35 KB | | |
| . 抽取式研羅 (F:) | img51.jpeg | 2016/2/2 上午 12 | JPEG 影像 | 35 KB | | |
| | img52.jpeg | 2016/2/2 上午 12 | JPEG 影像 | 35 KB | | |
| 42.92 | 🔊 img53.jpeg | 2016/2/2 上午 12 | JPEG 影像 | 35 KB | | |
| 2420 | 🔛 img54.jpeg | 2016/2/2 上午 12 | JPEG 影像 | 35 KB | | |
| | 📔 img55.jpeg | 2016/2/2 上午 12 | JPEG 影像 | 35 KB | | |
| | 🔛 img56.jpeg | 2016/2/2 上午 12 | JPEG 影像 | 35 KB | | |
| | 🔛 img57.jpeg | 2016/2/2 上午 12 | JPEG 影像 | 35 KB | | |
| | 🚳 lapse.bat | 2016/11/1 下午 0 | Windows 批次檔案 | 1 KB | | |
| | 🔊 output.mp4 | 2017/1/23 上午 1 | MP4 視訊 | 328 KB | | |



2.8 Media Single Stream – Source RTP Stream, Sink I2S

Please refer to following steps to understand how to run the MMF example with source as RTP audio stream and sink as I2S.

Pre-requisites

- Ameba board
- ALC5651 Ext. board
- Earphone
- USB cable is used to connect with PC.
- WiFi connection for receiving RTP streaming
- VLC media player on the testing PC

Procedures

- Please set "CONFIG_EXAMPLE_MEDIA_AUDIO_FROM_RTP" to be "1" in "platform_opts.h". The example code can be found in the project path, utilities → example → example_media_audio_from_rtp.c.
- In order to achieve the better audio quality but not required, please try to modify the following definitions for improving the WLAN performance for RTP packets. The following is an example.

In lwipopts.h,

PBUF_POOL_SIZE 20 => 40 PBUF_POOL_BUFSIZE 500 => 250 DEFAULT_UDP_RECVMBOX_SIZE 6 => 24

In opt.h,

MEMP_NUM_NETBUF 2 => 24

- Please compile the project and download the application into Ameba board as described in the previous section.
- Next, please connect the ALC5651 Ext. board with Ameba board. Then, please also connect the earphone to audio-out, labelled as "PH2", on ALC5651 Ext. board.
- As described in the previous section, please connect both the testing PC and Ameba to the same WiFi AP. As a result, the following figure can be shown. While the log message, "wlan mode:2", is shown in the log console, it means that the RTP server is ready.





 Next, please open the VLC media player, click "Media→Stream..." option in menu bar, and add the WAV file (G711 only) as the bellow figure. Then, click "Stream".

| 🛓 Open Media | |
|---|-----------------|
| File Disc n Network Capture Device File Selection You can select local files with the following list and buttons. | |
| D:\lucien.zhliu\Desktop\g711-ulaw-25s.wav | Add Remove |
| Use a subtitle file | Browse |
| Show more options | Stream v Cancel |

• In the window as the following figure, please click "Next" and, in the next window, set the "New destination" to be "RTP Audio/Video Profile" and click "Add".

| Stream Output | 2 × | 📥 s | Stream Output |
|---|---|-----|---|
| Cource Set up media sources to stream | | De | estination Setup Select destantions to stream to |
| This wizard will allow you to stream or convert your media for use locally, on your private ne You should start by checking that source matches what you want your input to be and then po | twork, or on the Internet. 88 the "Next" button to continue. | | • |
| | | | Add definations following the streaming methods you need. Be sure to check with transcoding that the format is compatible with the method used. |
| Source: file //ID /laxies.mbissDeukkop/g/111-ukev-25x.vev Type: file | | | New destantion ETP Audio/Yideo Profile Add Display locally |
| | Back Next Cancel | _ | Bick Next Ceacel |

• Please refer to the following figure and type the Ameba board's IP address and the RTP port (the default value is 16384). Then, click "Next".





| tination Setu elect destinatio | p ns to stream to | | | | |
|--|-----------------------------|--------------------|--------------|--|--|
| • | R TP/A V P 🔀 | | | | |
| This module o | outputs the transcode | d stream to a netw | ork via RTP. | | |
| | | | | | |
| Address Base port | 192.168.0.101 16384 | | | | |
| Address Base port Stream name | 192.168.0.101 | | | | |

• In this step, please select the profile as shown in the following figure and click "Next".

| 🛓 Stream Output | 2 × |
|--|------------------|
| Transcoding Options Select and choose transcoding options | |
| V Activate Transcoding | |
| Profile | Audio - CD 💉 🕅 |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | Back Next Cancel |

• Finally, the VLC generated stream output string is the same as that in the bellow figure, and click "Stream" to start the RTP transmission. Please also check the earphone plugged in the ALC5651 Ext. board whether the voice is out or not.

| A Stream Output | ? <mark>×</mark> |
|---|--------------------|
| Option Setup Set up any additional options for streaming | |
| Missellaneous Options | |
| C Stream all elementary streams | |
| Generated stream output string | |
| | |
| E | lack Stream Cancel |

2.9 Media Multiple Streams – Source GEO Camera, Sink RTP Stream

Please refer to following steps to understand how to run the MMF example with source as GEO camera and sink as RTSP stream.

Pre-requisites

- Ameba board
- GEO camera



- USB cable is used to connect with PC.
- WiFi connection for sending RTSP streaming out
- USB OTG cable is used to connect GEO camera to Ameba board.

Procedures

- As the Hardware Setup is presented in Section 3.1, please connect GEO camera, instead of UVC camera, to the Ameba board.
- Please set "CONFIG_EXAMPLE_MEDIA_MS" to be "1" in "platform_opts.h". The example code can be found in the project path, utilities → example → example_media_ms.c.
- In order to achieve the better streaming quality but not required, please try to modify the following definition for improving the WLAN performance for RTSP packets. The following is an example.

In rtw_opt_skbuf.c,

MAX_SKB_BUF_NUM 8 => 12

- Please compile the project and download the application into Ameba board as described in the previous section.
- As described in the previous section, please connect both the testing PC and Ameba to the same WiFi AP. As a result, the following figure can be shown. While the log message, "Streamon successful", is shown in the log console, it means that the RTSP server is



 Next, please open the VLC media player, click "Media→Open Network Stream" option in menu bar, and type in: "rtsp://xxx.xxx.xxx:yyy/test.sdp" before clicking play button.

xxx.xxx.xxx.xxx: the Ameba IP address.

yyy: RTSP server port number (default is 554).

• Please check whether both video and audio work normally or not by the VLC media player, in the testing PC, as shown in the figure as bellow.





2.10Media Multiple Streams – Source GEO Camera, Sink SD Card

Please refer to following steps to understand how to run the MMF example with source as GEO camera and sink as SD card.

Pre-requisites

- Ameba board
- GEO camera
- USB cable is used to connect with PC.
- SD card and SD/MMC card connector

Procedures

- Please refer to the document (UM0073 Realtek Ameba-1 Storage User Manual.pdf) to follow the steps in Section 2 to connect SD/MMC card connector with Ameba board. Then, please also plug SD card into SD/MMC card connector.
- Please set "CONFIG_EXAMPLE_MEDIA_GEO_MP4" to be "1" in "platform_opts.h". The example code can be found in the project path, utilities → example → example_media_geo_mp4.c.
- Please compile the project and download the application into Ameba board as described in the previous section.
- While the log message, "Streamon successful", is shown in the log console as the figure bellow, it is going to write mp4 file into SD card.





• Please wait a while then pull out the SD card from the SD card connector on the Ameba board. Then, please also connect SD card with another PC or laptop to check whether the mp4 files exist or not, as shown in the figure bellow, and whether both audio and video work normally by the VLC media player or not.

| | | | | * # = = = | | | × |
|--------------|-------------|----------------|--------|-------------|------------------------------------|---|---|
| | 抽取式磁碟 (F:) | | | | ◆ ◆ 援辱抽取式磁碟(F) | | Q |
| 組合管理 ▼ 共用對象 | ▼ 新埔資料夾 | | | | 8 | • | 0 |
| ☆ 我的最麗 | 名稱 | 修改日期 | 類型 | 大小 | | | |
| 📃 Desktop | AMEBA_0.mp4 | 2016/2/2 上牛 12 | MP4 視訊 | 29,453 KB | | | |
| 🚺 Downloads | AMEBA_1.mp4 | 2016/2/2 上午 12 | MP4 視訊 | 29,325 KB | | | |
| RecentPlaces | | | | | | | |
| | | | | | | | |
| [] 煤體櫃 | | | | | | | |
| | | | | | | | |
| ≣⊒ 1010 | | | | | | | |
| Cant Cant | | | | | | | |
| | | | | | | | |
| 馬 電腦 | | | | | | | |
| 🏭 WIN7 (C:) | | | | | | | |
| DATA (D:) | | | | | | | |
| MBED (E:) | | | | | | | |
| 👝 抽取式磁碟 (F:) | | | | | | | |
| | | | | | | | |
| 👊 網路 | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| 2 個項目 | | | | | | | |
| <i>~</i> | | | | | | | |
| | | | | | | | |

2.11MP3 Media Streaming Over RTP (Source-RTP; Sink Mp3 I2S).

This example highlights the streaming of mp3 files over wifi to the ameba and decoding of the mp3 locally in the ameba board and audio output from the audio jack. The example has two methods to provide audio output.

- Audio output using the ALC5651 board.
- Audio output using SGTL5000 board.



Pre-requisites

- Ameba board
- USB cable is used to connect with PC.
- ALC5651 Extension board or SGTL5000 Extension board.
- 3.5mm Audio jack to hear audio output.

Procedures

- In platform_opts.h edit the macro "CONFIG_EXAMPLE_MP3_STREAM_RTP" and set it to 1.
- By default the example uses the ALC5651 extension board is used for audio output but in case you wish to use the SGTL5000 audio extension board please set the macro "CONFIG_EXAMPLE_MP3_STREAM_SGTL5000" to one as well in the platform_opts.h file.
- The ALC5651 extension board fits like a shield on the AMEBA but in order to setup the SGTL5000 board please refer to :- <u>http://www.amebaiot.com/en/standard-sdk-i2s-audio-demo/</u>
- Once the macros are set please re-build the application project and flash the AMEBA board appropriately.
- Once flashed reset the AMEBA board and connect the board to wi-fi using the AT Commands and ensure the board gets an ip address as shown below.



• Once this is done, Open vlc media player on any other device connected to the same wifi network. Select the stream option(media->stream) as shown below.





• Once you select stream, use add and select the mp3 file that you wish to stream over rtp as shown below.

| Mec | File Disc Disc Capture Device Capture Device | |
|------|---|---|
| | File Selection | H |
| = 1 | You can select local files with the following list and buttons. | |
| My C | C:\Users\prashantravi\Downloads\sugar.mp3 | |
| | Remove | |
| | | |
| | | |
| | Use a subtitle file | |
| | Browse | |
| | | |
| | | |
| | | ⊢ |
| | Show more options | Ľ |
| | | |

Once you select the mp3 file click the stream option and proceed to the next window.
 Click next again and till you reach the "Destination Support" tab and select the destination as "RTP Audio/Video Profile" as shown below and click "Add"

| 🛓 s | Stream Outp | ut | | ? × |
|-----|------------------------------|--|--|----------|
| De | stination Select destination | etup ations to stream to | | |
| | ÷ | | | |
| | Add destin method use | ations following the streaming methods you ne cd. | red. Be sure to check with transcoding that the format is compatible t | with the |
| | New destin | ation | RTP Audio/Video Profile | Add |
| | Display | locally | | |
| _ | | | | |
| | | | Back Next | Cancel |

• Once in add, please enter the ip address of the ameba board that was assigned when you connected to the wifi hotspot. Enter the base port address as: - 16384 as shown below.

| 🛓 VLC r | media player | | | | | | × | | / |
|---------|-------------------------------------|------------------------|----------------|-----------------|------|------|------|--------|---|
| 🛓 S | tream Output | | | | | | | ? | x |
| Der | stination Setu Select destinatio | IP ons to stream to | | | | | | | |
| M | ф ; | RTP/AVP 🔀 | | | | | | | |
| | This module ou | utputs the transc | oded stream to | a network via l | RTP. | | | | |
| Ċ | Address | 192.168.0.54 | | | | | | | |
| | Base port | 16384 | | | | | | | |
| | Stream name | | | | | | | | |
| | | | | | | Back | Next | Cancel | |



• Click next and select the streaming option "Video – H.264+MP3(MP4)" after this click the settings button just beside the dropdown menu as shown below.

| <u> </u> | VLC media player | | |
|----------|--|---------------------------|-----------------------|
| ľ | 🛓 Stream Output | | ? × |
| | Transcoding Options Select and choose transcoding options | | |
| M | Activate Transcoding | | |
| | Profile | Video - H.264 + MP3 (MP4) | - 🖹 🗶 💼 |
| | | | Edit selected profile |
| | | | |
| | | | |
| | | | |
| | | | |
| Ľ | | | |
| | | | |
| | | | |
| | | Back | Next Cancel |

• Once in the settings option MP3 from the checkboxes as shown below.

| 4 | VLC media p | layer | | | _ _ × | |
|---|----------------------|--------------------|----------------------|-----------------|--------------|---------------|
| Y | A Stream | 🛓 Profile edition | | | | 2 × 2 × |
| P | Transcod Select a | Profile Name Video | - H. 264 + MP3 (MP4) | | | |
| = | | Encapsulation | Video codec Audio | codec Subtitles | | |
| M | 🗖 Acti | | | | | |
| | Profile | MPEG-TS | 🔘 Webm | Ogg/Ogm | MP4/MOV | |
| | | MPEG-PS | MJPEG | © wav | O FLV | |
| | | MPEG 1 | MKV | © RAW | IVA (| |
| | | ASF/WMV | C FLAC | MP3 | | |
| Ċ | | Features | | | | |
| ٤ | | 🗙 Video | × | Subtitles | 🗙 Streamable | |
| | | 🖌 Audio | × | Menus | 🗙 Chapters | |
| | | | | | | |
| | | | | | Save | Cancel Cancel |

• Once the mp3 option is selected, go to the "Audio Codec" tab and select the checkbox "Keep original audio track" as shown below.

| 🛓 VLC media p | ayer | | |
|----------------------------|---|--------------------|---|
| A Stream | A Profile edition | <u> </u> | X |
| Pl Transcod Select a | Profile Name Video - H. 264 + MP3 (MP4) | | |
| el | Encapsulation Video codec Audio codec Subtitles | | _ |
| Acti | V Audio | | |
| Profile | Keep original audio track Encoding parameters Filters | | |
| | Codec MPEG Audio | • | |
| | Bitrate | 128 kb/s 🌲 | |
| | Channels | 2 🚖 | |
| | Sample Rate 44100 Hz | | |
| | | | |
| | | | |
| ſ | | | |
| | | | |
| | | | |
| | | | |
| | | Save Cancel Cancel | |
| | | | |



- Once the audio codec is set, click save, then click next and click "stream"
- Once the streaming begins in vlc media player, connect the audio jack to the AMEBA in order to hear the audio output in the earphones. Both AMEBA and the device where vlc media player is streaming should be connected to the same Wi-Fi network.
- To ensure smooth playback with better RTP performance, try utilizing the following options as shown below.

 \rightarrow in lwipopts.h,

PBUF_POOL_SIZE 20 => 40 (just an example) PBUF_POOL_BUFSIZE 500 => 250 DEFAULT_UDP_RECVMBOX_SIZE 6 => 24 (just an example) MEMP_NUM_NETBUF 2 => 24 (just an example) TCPIP_MBOX_SIZE 6 => 6 (just an example)

```
\rightarrowin opt.h,
```

2.12MP3 Play Audio from SDCARD

This highlights the use of the mp3 example for playing the mp3 media from SDCARD.

Pre-requisites

- Ameba board
- USB cable is used to connect with PC.
- ALC5651 Extension board or SGTL5000 Extension board.
- 3.5mm Audio jack to hear audio output.
- SDCARD sniffer in order to read SDCARD.

Procedures

- In platform_opts.h edit the macro "CONFIG_EXAMPLE_AUDIO_MP3" and set it to 1.
- Set the Config parameters in the beginning of the example file as shown below.
- The "I2S_DMA_PAGE_SIZE", "NUM_CHANNELS" and "SAMPLING_FREQ" also need to be set appropriately.

| | squancy mapping table | | |
|---------------------------------|--|---|--|
| Frequency(hz) Numi (CH_i | ber of Channels MONO:1 CH_STEREO:2) | Decoded Bytes (125_DMA_PAGE_SIZE | + E) |
| 8000 | 1 | 115 | + 52 |
| 8000 | 2 | 230 | 04 |
| 16000 | 1 | 115 | 52 |
| 16000 | 2 | 230 | 04 |
| 22050 | 1 | 115 | 52 |
| 22050 | 2 | 230 | 04 |
| 24000 | 1 | 115 | 52 |
| 24000 | 2 | 230 | 04 |
| 32000 | 1 | 230 | 04 |
| 32000 | 2 | 460 | 08 |
| 44100 | 1 | 230 | 04 |
| 44100 | 2 | 460 | 08 |
| 48000 | 1 | 230 | 04 |
| 48000 | 2 | 460 | 08 |
| + | | + | -+*/ |
| // | | -CONFIG Paramotone | |
| define I2S DMA PAGE : | SIZE 4608 //Use free | quency mapping tabl | le and set this value to number of decoded byt. |
| | //Options | - 1152, 2304, 4608 | 8 |
| define NUM_CHANNELS (| CH_STEREO //Use mp3 | file properties to | o determine number of channels |
| - | //Options | - CH MONO, CH STEP | REO |
| define SAMPLING FREQ | SR 32KHZ //Use mp3 | file properties to | o identify frequency and use appropriate macro |
| | //Options | - SR 8KHZ =>80 | 000hz - PASS |
| | 11 | SR 16KHZ =>16 | 6000hz - PASS |
| | 11 | SR 24KHZ =>24 | 4000hz - PASS |
| | | | |
| | 11 | SR 32KHZ =>32 | 2000hz - PASS |
| | | SR_32KHZ =>32 SR_48KHZ =>48 | 2000hz - FASS 8000hz - FASS |
| | | SR_32KHZ =>32 SR_48KHZ =>48 SR_96KHZ =>96 | 2000hz - FASS 8000hz - FASS 6000hz ~ NOT SUPPORTED |



- As shown above the I2S_DMA_PAGE_SIZE should be set to the value of decoded bytes for the particular frequency and number of channels using the frequency mapping table that is mentioned in section 2.4.2 and also provided in the c file as comments.
- Rebuild the project and download and flash the binary onto the Ameba board.
- Once the software is downloaded, connect the SDCARD sniffer and the audio output board to the appropriate pins on the Ameba board.
- Load the mp3 file onto the SDCARD and rename the file as "AudioSDTest.mp3" this is very essential for the mp3 file to be detected properly by the software.
- Once the SDCARD is ready, insert it into the SD sniffer and press the reset button, the audio should start playing from the audio jack.