AN13208

Hardware and software design using i.MX RT106S MCU based solution for Local Voice Control for IoT

Rev. 0 — 19 April 2021 Application Note

1 Introduction

The NXP i.MX RT106S MCU-based solution for Local Voice Control for IoT is a comprehensive and cost optimized solution. With its easy to use SLN-LOCAL2-IOT development kit, the solution enables device makers to quickly market Local Voice Control built-in designs based on the NXP end-to-end software application which is production ready, fully tested, and certified.

The SLN-LOCAL2-IOT development kit is based on the i.MX RT106S microcontroller powered by an Arm® Cortex®-M7 core.

NOTE

The SLN-LOCAL2-IOT development kit does not represent a recommended hardware implementation for i.MX RT106S based Local Voice Control for IoT designs.

The purpose of this application note is to:

- Guide developers through the process of implementing NXP recommended reference design.
- Enable easy migration of any software developed on the SLN-LOCAL2-IOT development kit to any production hardware based on this reference design.

The key differences between the SLN-LOCAL2-IOT development kit hardware and the recommended reference design include:

- Replacing the Murata 1DX Wi-Fi/BLE module with a similar AzureWave AW-NM372SM module.
- Replacing HyperFlash with Quad SPI NOR Flash.
- Replacing the Goodix (formerly NXP) smart amplifier TFA9894D with a digital to analog converter (“MQS”) integrated on the i.MX RT106S MCU and a low-cost analog audio amplifier.

There are two versions of the i.MX RT106S Local Voice Control for IoT software package downloadable through MCUXpresso SDK Builder:

- **SLN-LOCAL2-IOT** is the SDK for the SLN-LOCAL2-IOT development kit.
- **SLN-LOCAL2-RD** is the SDK for use with the developer’s hardware based on reference design from NXP.

Contents

2 Hardware reference design

Figure 1 shows the hardware architecture of the SLN-LOCAL2-IOT development kit.

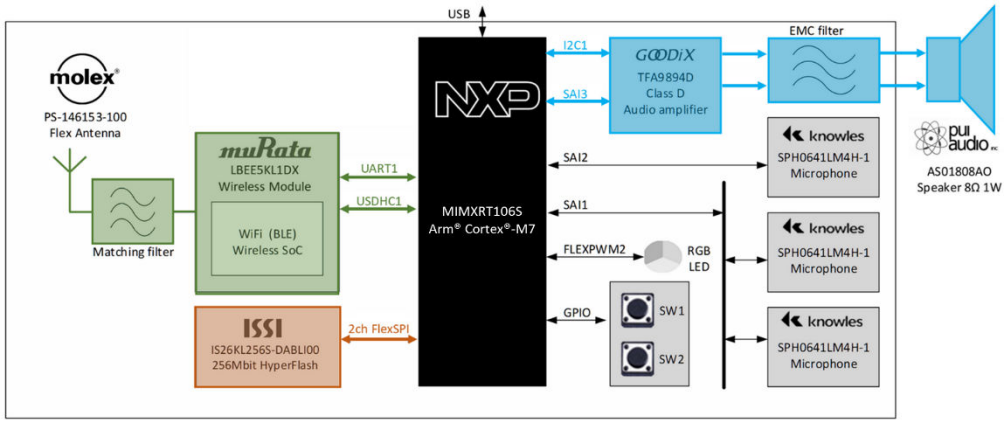

Figure 1. SLN-LOCAL2-IOT development kit hardware block diagram

Figure 2 shows the hardware architecture of the i.MX RT106S reference design.

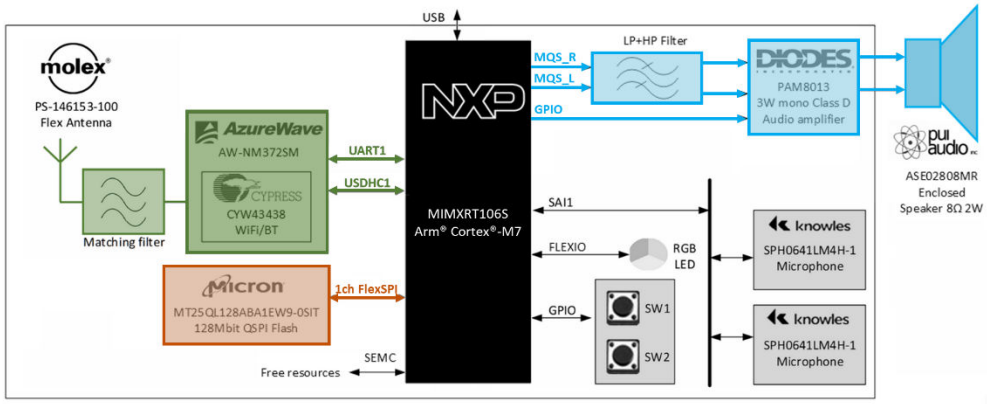

Figure 2. i.MX RT106S reference design hardware block diagram

2.1 Wi-Fi and Bluetooth module

i.MX RT106S MCU requires two physical interfaces to control and communicate with the Radio module.

- **SDIO 2.0 interface** for Wi-Fi IEEE802.11 b/g/n, supported by RT106S USDHC1 peripheral.
- **UART interface** for Bluetooth Low Energy (BLE), supported by RT106S UART1 module.

2.1.1 AzureWave AW-NM372SM module

The AzureWave AW-NM372SM module embeds a Cypress CYW43438_A1 combo chipset supporting Wi-Fi IEEE 802.11b/g/n and Bluetooth 4.2. This module features a 12.0 mm x 12.0 mm LGA package.

For requesting the AzureWave AW-NM372SM module datasheet, visit www.azurewave.com.

NOTE

The AzureWave AW-NM372SM module also supports an FM radio data receiver (RDS & RDBS). However, that functionality is not considered in this document.

2.1.2 Wi-Fi module schematics

Figure 3 details the schematics for the Wi-Fi module.

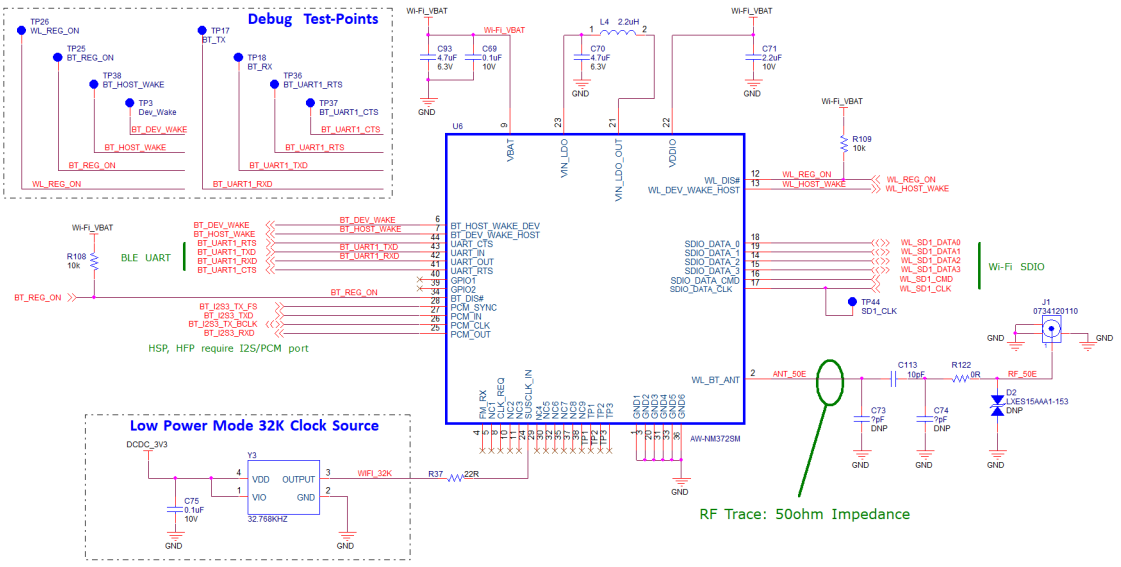

Figure 3. i.MX RT106S schematics with AzureWave AW-NM372SM module

2.1.3 Antenna line implementation

The AzureWave AW-NM372SM module features a radio input/output pin (WL_BT_ANT), which is connected to a 2.4 GHz antenna. The design of this connection optimizes the RF power transfer from the antenna to the module input pin. Failure to do so limits the receiver sensitivity and affects the transmitter effective output power, leading to a reduced connection range for both Wi-Fi and Bluetooth Low Energy.

The key design considerations for any new development are:

- PCB trace must have a 50 Ω characteristic impedance in the range [2.4 GHz; 2.5 GHz].
- Pi-filter is required to adjust the impedance matching and therefore optimize the RX/TX power transfer. (C73, C113, C74; R122). The component values are specific to each PCB layout implementation.
- 2.4 GHz RF connector manages the connection to a distant antenna (J1).
- Optional ESD protection is considered if it does not affect the line impedance (D2).

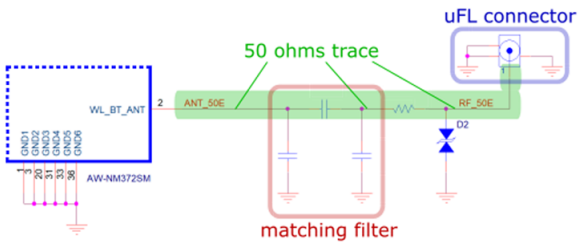

Figure 4. Antenna line

NOTE

The practical design of a 2.4 GHz antenna is beyond the scope of this document. For comprehensive information, see the application note AN91445 - Antenna Design and RF Layout Guidelines on the Cypress website.

2.1.4 Power supply line

The AzureWave AW-NM372SM module is powered through VBAT pin with following specifications:

- VBAT must be supplied with +3.3 V, to ensure compatibility with i.MX RT I/Os
- AzureWave AW-NM372SM module guarantees optimal RF performances with VBAT superior to +3.2 V
- VBAT supply must be designed to support +3.3 V +/- 3 %
- 3V3 power-supply design must tolerate 700 mA current peaks from the module

ATTENTION

It is recommended to incorporate an LC filter on the 3V3 supply line for the radio module. This filter helps in rejecting the noise injected in the supply line by digital components (i.MX RT, flash memory).

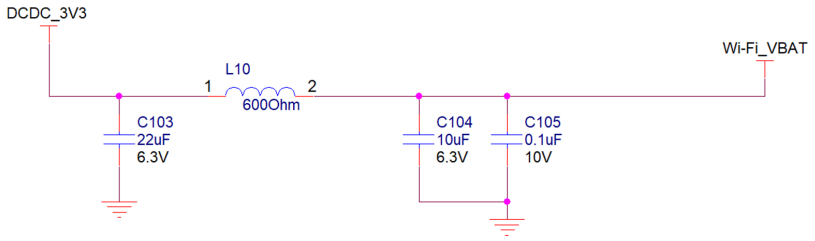

Figure 5. Recommended VBAT LC filter

2.1.5 Switching regulator filter

The Cypress CYW43438 chipset embeds a buck regulator. The regulator requires an LC filter (L4, C70) to operate properly. The loop VIN_LDO_OUT, LC filter, and VIN_LDO must be:

- Kept as short as possible to minimize its radiations
- Routed away from any other signal

ATTENTION

The current rating of the inductance L4 is critical. Thus, it is recommended to use the Murata LQM18PN2R2MGHD for that filter.

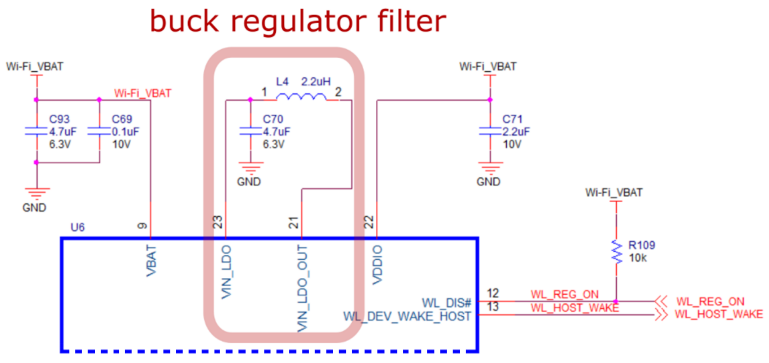

Figure 6. Buck regulator filter

2.1.6 UART interface

A four-wire UART interface connects the AzureWave AW-NM372SM module to the i.MX RT106S UART1 peripheral.

Table 1. UART Interface for Bluetooth/Bluetooth LE

Line	AW-NM372SM		i.MX RT106S		Description
	Pin	Type	Pin	Type	
BT_UART1_TXD	43. UART_IN	Input	K14. UART1_TXD	Output	AW-NM372SM UART Data in
BT_UART1_RXD	42. UART_OUT	Output	L14. UART1_RXD	Input	AW-NM372SM UART Data out
BT_UART1_RTS	44. UART_CTS	Input	L10. UART1_RTS	Output	AW-NM372SM UART Clear To Send
BT_UART1_CTS	41. UART_RTS	Output	H14. UART1_CTS	Input	AW-NM372SM UART Request To Send

This UART interface is dedicated to BT/BLE control/communication, it is 3V3 compliant and runs by default at 115200 bauds.

NOTE

Addition of some serial resistors on each UART line is recommended to facilitate Bluetooth /Bluetooth LE performance evaluation.

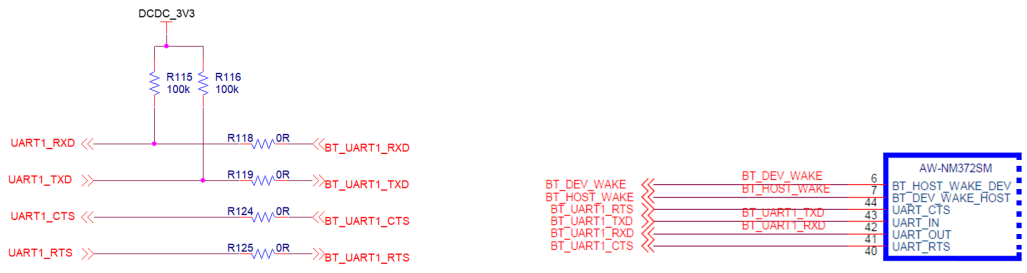

Figure 7. UART interface

Cypress via its Cypress Developer Community (https://community.cypress.com) is distributing a specific software utility called CyBlueTool to test the radio and measure the BT/BLE throughput.

2.1.7 SDIO interface

A six-wire SDIO interface connects the AzureWave AW-NM372SM module to the i.MX RT106S SDHC1 peripheral.

Table 2. SDIO interface for WLAN

Line	AW-NM372SM		i.MXRT		Description
	Pin	Type	Pin	Type	
WL_SD1_DATA0	18. SDIO_DATA_0	Input/Output	J1. SD1_DATA0	Input/Output	SDIO Data 0
WL_SD1_DATA1	19. SDIO_DATA_1	Input/Output	K1. SD1_DATA1	Input/Output	SDIO Data 1
WL_SD1_DATA2	14. SDIO_DATA_2	Input/Output	H2. SD1_DATA2	Input/Output	SDIO Data 2
WL_SD1_DATA3	15. SDIO_DATA_3	Input/Output	J2. SD1_DATA3	Input/Output	SDIO Data 3
WL_SD1_CMD	16. SDIO_DATA_CMD	Input	J4. SD1_CMD	Output	SDIO CMD

Table continues on the next page...

Table 2. SDIO interface for WLAN (continued)

Line	AW-NM372SM		i.MXRT		Description
	Pin	Type	Pin	Type	
WL_SD1_CLK	17. SDIO_DATA_CLK	Input	J3. SD1_CLK	Output	SDIO CLK
WL_SD1_DATA2	14. SDIO_DATA_2	Input/Output	H2. SD1_DATA2	Input/Output	SDIO Data 2
WL_SD1_DATA3	15. SDIO_DATA_3	Input/Output	J2. SD1_DATA3	Input/Output	SDIO Data 3
WL_SD1_CMD	16. SDIO_DATA_CMD	Input	J4. SD1_CMD	Output	SDIO CMD
WL_SD1_CLK	17. SDIO_DATA_CLK	Input	J3. SD1_CLK	Output	SDIO CLK

This SDIO interface is dedicated to Wi-Fi control/communication, it is 3V3 compliant and runs by default at 50 MHz.

NOTE

Addition of some pull-up resistors, bypass capacitors, and serial resistors on each SDIO line is recommended to facilitate the SDIO line matching.

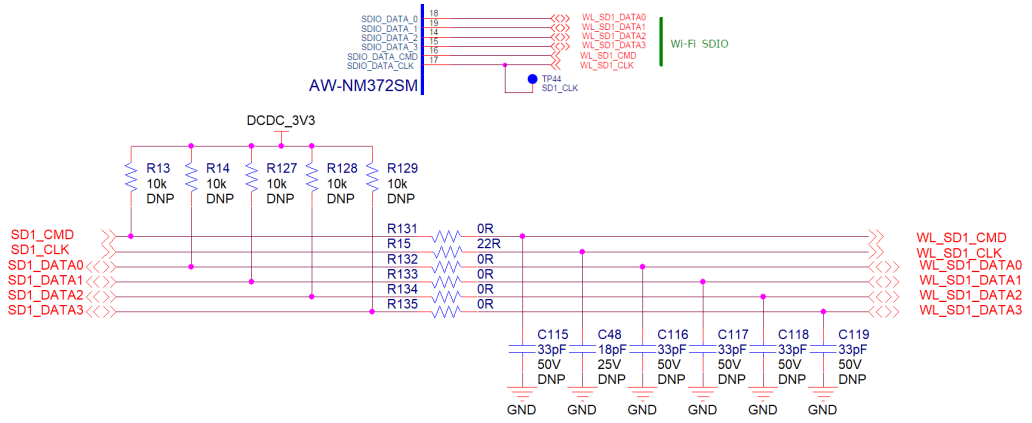

Figure 8. SDIO interface

2.1.8 Optional 32.768 kHz external oscillator

The schematics include an optional 32.768 kHz external oscillator Y3 connected to the AzureWave AW-NM372SM module. This oscillator is only required for low-power modes.

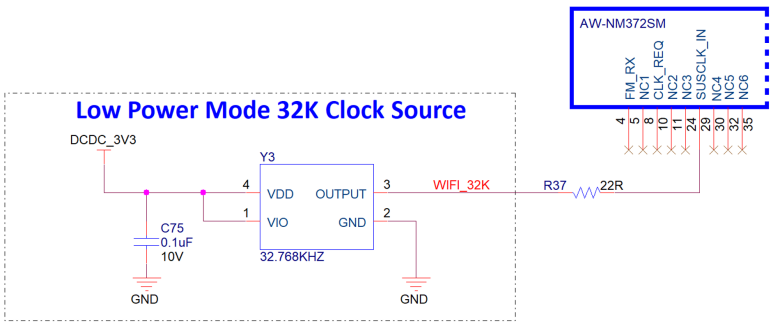

Figure 9. Optional 32.768 kHz external oscillator

2.1.9 GPIO controls

A set of standard GPIO controls the AzureWave AW-NM372SM module operation and power modes. The allocation in Table 3 is used and must be copied, as much as possible, to ensure compatibility with Local2 SDK solution software.

Table 3. GPIO control signals

Line	AW-NM372SM		i.MX RT		Description
	Pin	Type	Pin	Type	
WL_REG_ON	12. WL_DIS#	Input	K7. PMIC_ON_REQ	Output	Power up / down the WLAN regulator
WL_HOST_WAKE	13. WL_DEV_WAKE_HOST	Output	L6. WAKEUP	Input	i.MX RT wake-up signal
BT_REG_ON	34. BT_DIS#	Input	C14. GPIO_B1_14	Output	Power up / down the Bluetooth / Bluetooth LE regulator
BT_HOST_WAKE	7. BT_DEV_WAKE_HOST	Output	B14. GPIO_B1_15	Input	i.MX RT wake-up signal
BT_DEV_WAKE	6. BT_HOST_WAKE_DEV	Input	H1. GPIO_EMC_12	Output	Bluetooth wake-up signal

2.2 Flash memory

The i.MX RT106S MCU uses FlexSPI interface to control and communicate with the Flash module with:

- two channels for HyperFlash memories, as used on the SLN-LOCAL2-IOT development kit.
- one channel for Quad SPI NOR Flash, as used on the i.MX RT106S reference design.

2.2.1 Quad SPI NOR Flash minimum requirements

Use of Quad SPI NOR Flash memory with a minimum throughput of 60 MB/s is recommended to ensure proper execution of the Local Voice Control Application. A module with 128 Mbit capacity is sufficient to store the Local Voice Control Application, its file system, with room for an over the air (OTA) firmware update and other application software.

For power supplies, 3 V Flash modules allow simpler design, while 1.8 V memory lowers the board power consumption. Ball grid array (BGA) packages are preferred for compact applications while SO8 package are cheaper.

NXP used the Micron MT25QL128ABA1EW9-0SI to validate this document.

NOTE

QSPI NOR Flash modules with capacities up to 128 Mbit support 3 Byte address mode, while higher capacities require 4 Byte address mode.

2.2.2 Boot configuration settings

The i.MX RT106S MCU can boot from different types of flash module, such as QSPI NOR Flash, HyperFlash, Octal Flash, and even external SD-Card. The boot configuration is set in hardware by connecting i.MX RT106S BOOT_CFG[0-11] pins to GND or VDD as described in the Table 4.

Table 4. RT106S Boot configuration options

	0	0	1	1	0	0	0	0	0	0	0	0
TYPE	**BOOT_CFG[11]**	**BOOT_CFG[10]**	**BOOT_CFG[9]**	**BOOT_CFG[8]**	**BOOT_CFG[7]**	**BOOT_CFG[6]**	**BOOT_CFG[5]**	**BOOT_CFG[4]**	**BOOT_CFG[3]**	**BOOT_CFG[2]**	**BOOT_CFG[1]**	**BOOT_CFG[0]**
FlexSPI1 - Serial NOR	*Infinit-Loop: (Debug USE only) 0 - Disable 1- Enable*	*FLASH_TYPE 000-Device supports 3B read by default 001-Device supports 4B read by default 010-HyperFlash 1V8 011-HyperFlash 3V3 100-MXIC Octal DDR*			*0*	*0*	*0*	*0*	*HOLD TIME: 00 - 500us 01 - 1ms 10 - 3ms 11 - 10ms*		*EncryptedXIP 0 - Disabled 1- Enabled*	*Reserved*
SD	*Infinit-Loop: (Debug USE only) 0 - Disable 1- Enable*	*Reserved*	*Bus Width: 0 - 1-bit 1 - 4-bit*	*SD1 VOLTAGE SELECTION: 0 - 3.3V 1 - 1.8V*	*0*	*1*	*SD/SDXC Speed: 00 - Normal/SDR12 01 - High/SDR25 10 - SDR50 11 - SDR104*		*SD Power Cycle Enable: '0' - No power cycle '1' - Enabled via USDHC_RST pad*	*SD Loopback Clock Source Sel: (for SDR50 and SDR104 only) '0' - through SD '1' - direct*	*Port Select: 0 - eSDHC1 1 - eSDHC2*	*Fast Boot: 0 - Regular 1 - Fast Boot*

To support some 128 Mbit QSPI NOR Flash module with 3 Byte address mode, the i.MX RT106S hardware boot configuration is as shown in Figure 10.

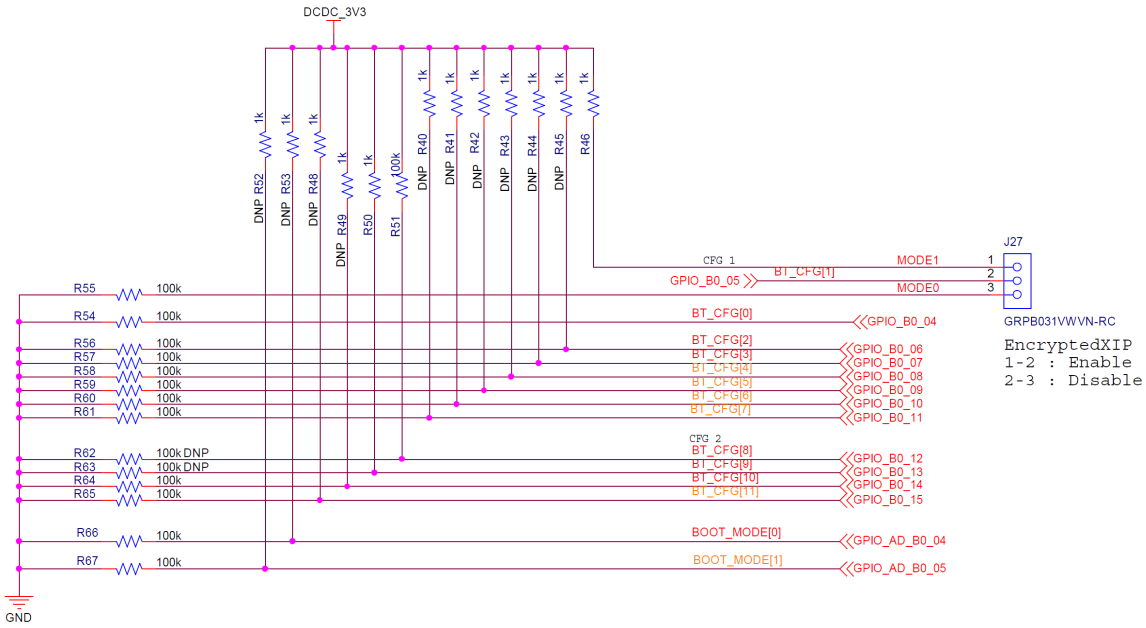

Figure 10. RT106S boot configuration for Quad SPI NOR Flash

2.2.3 Flash schematics

Figure 11 details the Flash memory schematics for the i.MX RT106S.

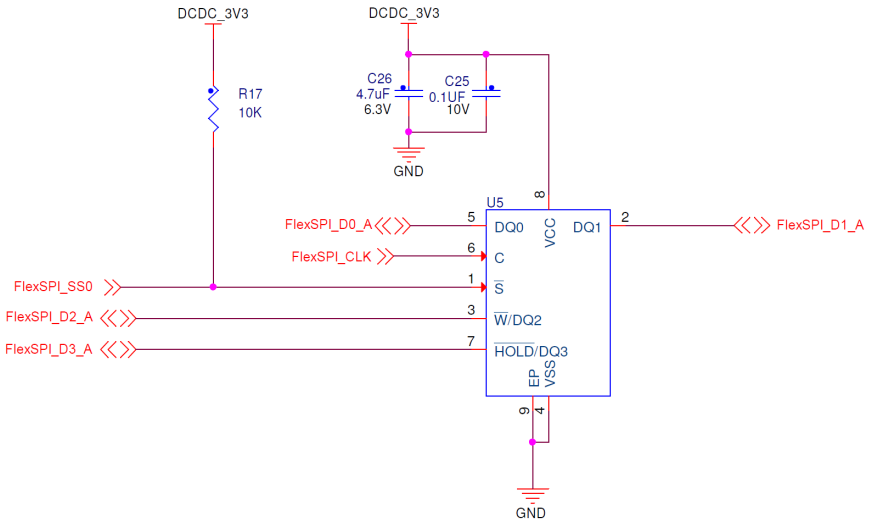

Figure 11. Schematics with Quad SPI NOR Flash module

2.2.4 i.MX RT DQS Pin

Most of the QSPI NOR Flash memory available on the market do not feature a DQS pin, unlike HyperFlash™ memory or Octal Flash memory, which is required to realign data signals at high frequency.

i.MX RT106S FlexSPI controller supports QSPI NOR Flash memory without DQS pin up to 133 MHz in SDR mode and 60 MHz in DDR mode.

i.MX RT106S DQS pin must be left floating and the targeted FlexSPI controller must be configured in software to use a Loopback from DQS Pad.

In the solution, the QSPI NOR Flash memory is connected to i.MX RT106S FlexSPI_A controller. According to the *Table 8.1 - ROM Bootloader Peripheral PinMux* from the *i.MX RT1060 Reference Manual*, the default pin for FLEXSPI_A_DQS verified by i.MX RT106S Boot-ROM is GPIO_SD_B1_05 (N3). The default pin must be left floating as shown in Figure 12.

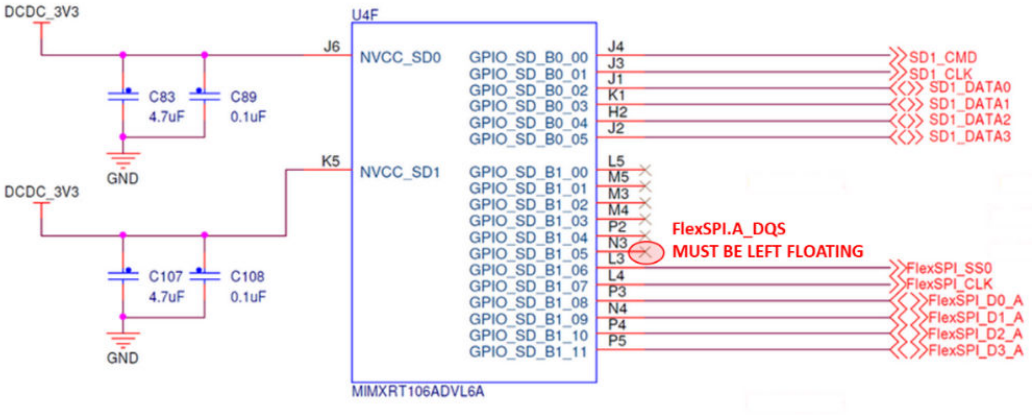

Figure 12. i.MX RT106S DQS pin not connected

2.2.5 Reducing current and electromagnetic radiations

Selecting the optimum pin/port configuration reduces the current draw and the electromagnetic radiations.

For more information on the slew rate and drive strength pin configuration, see the application note titled *Influence of Pin Setting on System Function and Performance* (document: AN5078).

For more information on the Spread Spectrum pin configuration, see the application note titled *How to Enable Spread Spectrum for RT Family* (document: AN12879).

2.3 Audio amplifier

On the SLN-LOCAL2-IOT development kit, three physical interfaces are required for the i.MX RT106S MCU to control and communicate with the Goodix TFA9894DUK Smart Audio Amplifier.

The requirements are:

- **I2S/SAI interface** with Tx_Data, Tx_Sync, Tx_BCLK signals to transmit the audio data and Rx_Data signal for the feedback loop (barging requirement).
- **I2C interface** to set up and control the TFA9894 Smart audio amplifier.
- **Two GPIO signals** to reset the TFA9894 and send interrupts to the i.MXRT106S MCU.

On the i.MX RT106S reference design two physical interfaces are required to control and communicate with the analog audio amplifier using the integrated MQS controller.

The requirements are:

- **Two MQS ports** left and right to transmit the audio signal to the audio amplifier.
- **One GPIO signal** to shut down the analog audio amplifier and minimize drawn current in idle mode.

The MQS controller is used to generate medium quality audio via standard digital GPIO pins allowing the user to connect stereo speakers or headphones to a power amplifier without an additional DAC chip.

The 44 kHz or 48 kHz input signals from SAI3 are in left_justified format. MQS provides the SNR target as no more than 20 dB for the signals below 10 kHz. The signals above 10 kHz have worse THD+N values.

MQS provides only simple audio reproduction. No internal pop, click, or distortion artifact reduction methods are provided.

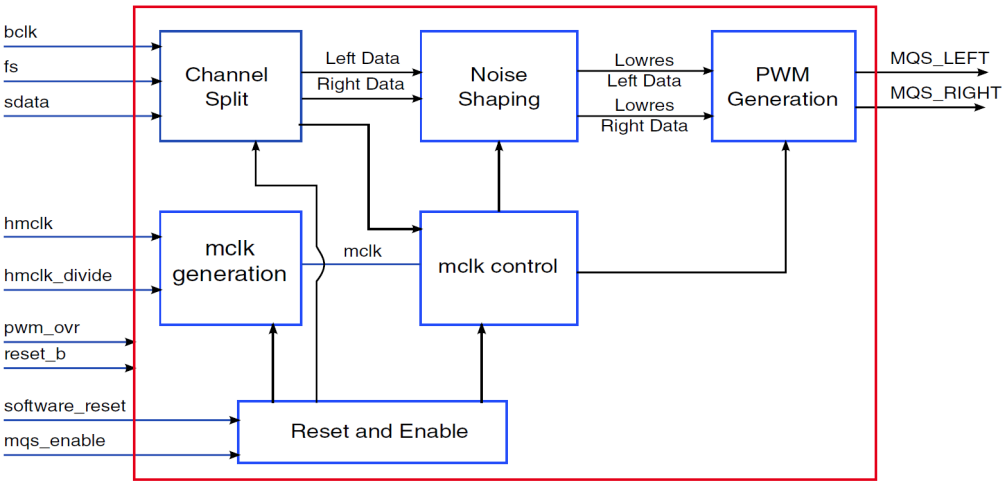

Figure 13. MQS controller block diagram

2.3.1 Analog Audio Amplifier minimum requirements

Use of a Class-D amplifier for a higher efficiency, with differential input and high PSRR is recommended to eliminate noise and RF rectification.

The reference design uses the Diodes Inc. PAM8013 Class-D mono 3 W Analog Audio Amplifier and the PUI ASE02808MR-LW150-R speaker to validate this document.

Two complementary digital signals are generated and are fed to the LEFT and RIGHT outputs of the MQS controller to drive the analog amplifier in a differential way from a MONO audio input.

Low-cost analog audio amplifiers do not support any gain control feature. Thee combination of the PAM8013 internal feedback and the resistor R47/R48 values determine the main audio gain control. This function can also be implemented in software. One basic implementation is a bit shift of the PCM signal before the MQS controller input. For details, see the Software Patch.

R44/C28 and R43/C27 forms a first-order low-pass filter that removes part of the high frequency component (clock frequency) of the PWM signal, which is out of the audio frequency band.

The ferrite beads FB3/FB4 reduce EMI around 1 MHz and higher

NOTE

Load and power parameter from the speaker targeted with 2 W at 8 Ohms must be powerful enough for most voice applications.

.

2.3.2 Audio schematics

Figure 14 details the audio schematics.

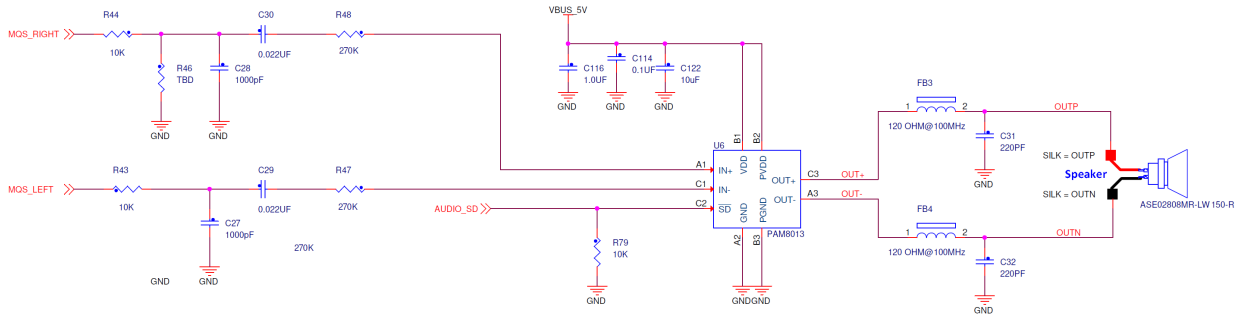

Figure 14. Analog audio amplifier schematics

The i.MX RT106S MCU can output AUDIO_P and AUDIO_N signals on several pins. However, use of GPIO_B0_00 and GPIO_B0_01 is recommended to reserve the other options for the boot configuration or a potential external SDRAM upgrade.

Table 5. i.MX RT106S MQS pin options

GPIO Name	Signal Alt1	Signal Alt2	Comment
GPIO_AD_B0_04	MQS_RIGHT	BOOT_MODE0	Not Recommended
GPIO_B0_00	MQS_RIGHT	NC on SLN-LOCAL2-IOT	**Recommended**
GPIO_EMC_13	MQS_RIGHT	SEMC_A4	Reserve for SDRAM Memory
GPIO_EMC_14	MQS_LEFT	SEMC_A5	Reserve for SDRAM Memory

Table continues on the next page...

Table 5. i.MX RT106S MQS pin options (continued)

GPIO Name	Signal Alt1	Signal Alt2	Comment
GPIO_B0_01	MQS_LEFT	NC on SLN-LOCAL2-IOT	**Recommended**
GPIO_AD_B0_05	MQS_LEFT	BOOT_MODE1	Not Recommended

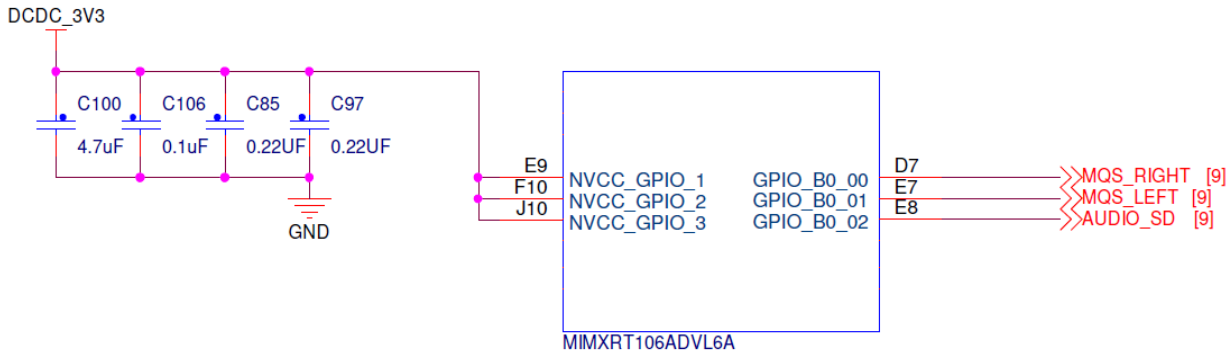

Figure 15. i.MX RT106S MQS outputs connected to the analog audio amplifier

3 i.MX RT106S Local Voice Control for IoT software

This section provides an overview of the steps required to install SDK package, SDK contents, Wi-Fi and Bluetooth module (AzureWave AW-NM372SM), Quad SPI NOR Flash memory (Micron MT25QL128ABA), Analog Audio Amplifier with MQS (Diode Inc. PAM8302A), and other software updates.

3.1 SDK package installation

The SLN-LOCAL2-RD SDK package version 2.8.0, which can be downloaded from https://mcuxpresso.nxp.com/en/selectsupports the i.MX RT106S reference design.

The SDK package contains multiple projects to help evaluate the key Local Voice components. This document describes the **sln_voice_examples** project and focuses specifically on the **local_demo** project. However, the projects differ from the projects in the SLN-LOCAL2-IOT SDK package.

The SDK project structure is shown in Figure 16.

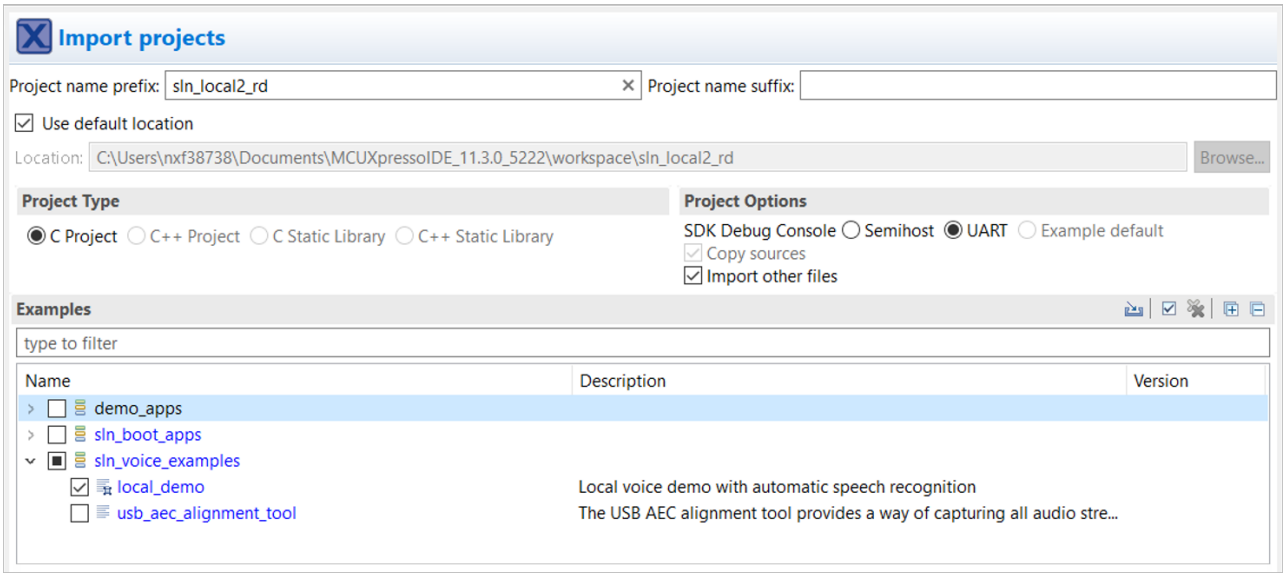

Figure 16. Project examples

For accessing these projects, open MCUXpresso IDE and then select the view **Installed SDKs**. Drag and drop the SLN-LOCAL2-RD zip package in this view.

If the SLN-LOCAL2-RD SDK is linked to an archive (ZIP file) or a folder as shown Figure 17, the SDK is imported successfully.

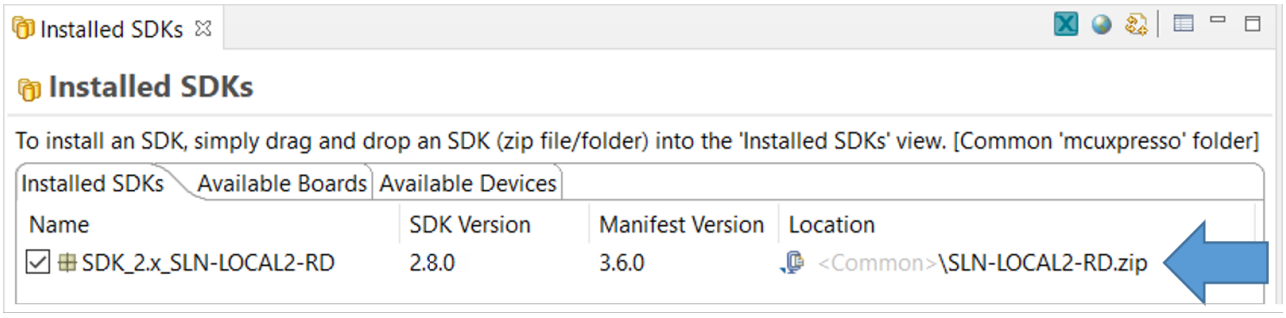

Figure 17. SDK installation format

3.2 SLN-LOCAL2-RD SDK content description

This SDK is adding support for the AzureWave AW-NM372SM Wi-Fi and Bluetooth module (embedding the Cypress CYW43438 chipset), 128 Mbit QSPI NOR Flash (using the Micron MT25QL128ABA) and an Analog Audio Amplifier with MQS (using the Diode Inc. PAM8302A).

Each part support will be detailed in its own section.

3.2.1 Wi-Fi and Bluetooth module (AzureWave AW-NM372SM)

Figure 18 shows the different folders and files for the reference design Wi-Fi and Bluetooth module.

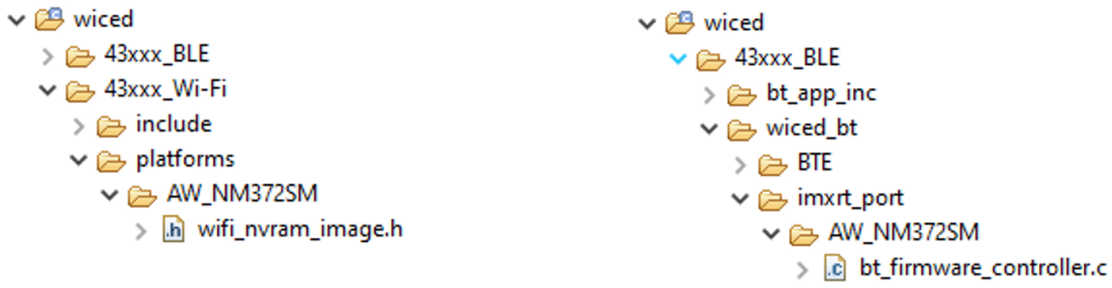

Figure 18. Modified files for Wi-Fi and Bluetooth module

The new SDK changes the RF parameters configuration files stored in NVRAM. The parameters are specific for the Wi-Fi and Bluetooth LE modules and are provided by the supplier for best performance and FCC standard compliance.

3.2.2 Quad SPI NOR Flash memory (Micron MT25QL128ABA)

Figure 19 shows the different folders/files for the reference design QSPI NOR Flash memory.

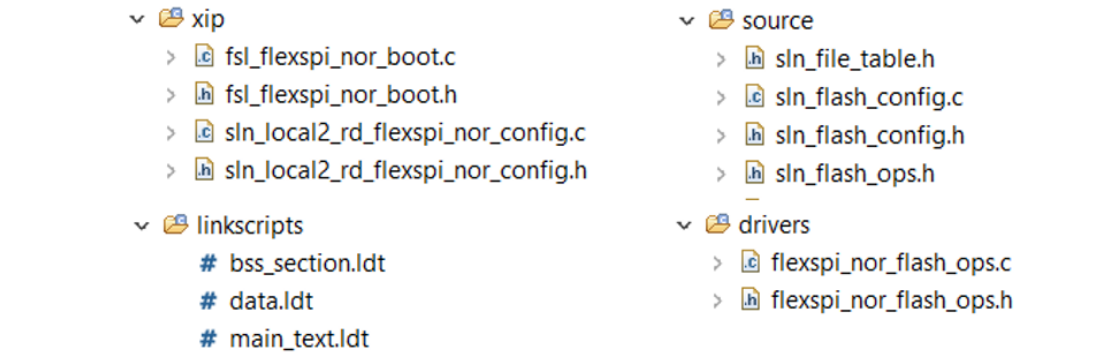

Figure 19. Modified files for QSPI NOR Flash memory

The main changes in the new SDK include:

- Updated Flash LUT Table that contains the proper commands for reading, writing, erasing the flash and more.
- Updated flash map since the addresses at which we keep our files have changed as well, due to the smaller sectors 4 KB vs 256 KB.
- Updated flash drivers that use the QSPI LUT table instead of the HyperFlash one.
- Linkscripts **data.ldt** and **main_text.ldt** now refer to QSPI objects.

These changes are for setting up and using the QSPI Flash chip. Therefore, they must not be modified unless there is a specific reason to do it. If modified, the projects may generate unexpected behavior, errors, or stop working altogether.

3.2.3 Analog Audio Amplifier with MQS (Diode Inc. PAM8302A)

Figure 20 shows the different folders/files for the reference design Analog Audio Amplifier with MQS.

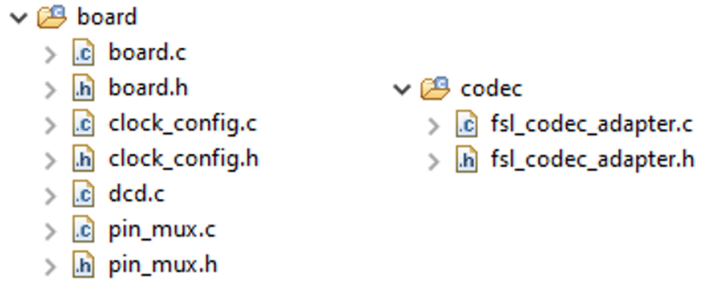

Figure 20. Modified files for MQS support

USE_MQS definition in project settings is set to 1, while **USE_TFA** definition is deleted. This enables some **MQS**-specific code in pdm_to_pcm_task.*, sln_amplifier.*, sln_pdm_mic.*, pdm_pcm_definitions.h, board.c, and main.c, while some **TFA**-specific code is disabled.

The change in **pin mux files** initializes the GPIOs and ensures connectivity to the **PAM8302A** amplifier, while the change in **board.c** initializes the **PAM8302A** and removes the receive DMA handle. The **PAM8302A** amplifier does not communicate back with the board and therefore the audio data flow changes.

The **TFA** amplifier used on the development kit has two main features that not available on the **PAM8302A** amplifier used on the reference design. The features are the volume control and the feedback loop, which is used for the barge-in (remove sound generated by the speaker from the signal received by the microphones).

To address these missing features, the **sln_amplifier.c** file is updated to achieve these two features in the software by adding one internal function and replacing another.

- **SLN_AMP_VolAndDiffInputControl(int16_t *data, uint32_t len)**: Deals with the volume change by shifting the audio data by x amount to the right. This also prepares the audio data for being sent to the amplifier in differential mode.
- **SLN_AMP_RxCallback(uint8_t *data, uint32_t length)**: Is replaced to send the audio data received from the streamer directly to the pdm_to_pcm_task.

At the same time, the common codec is updated for the MQS (fsl_codec_adapter.h API) and all TFA codec-related files are removed.

The audio data flow changes in the reference design as shown in Figure 21.

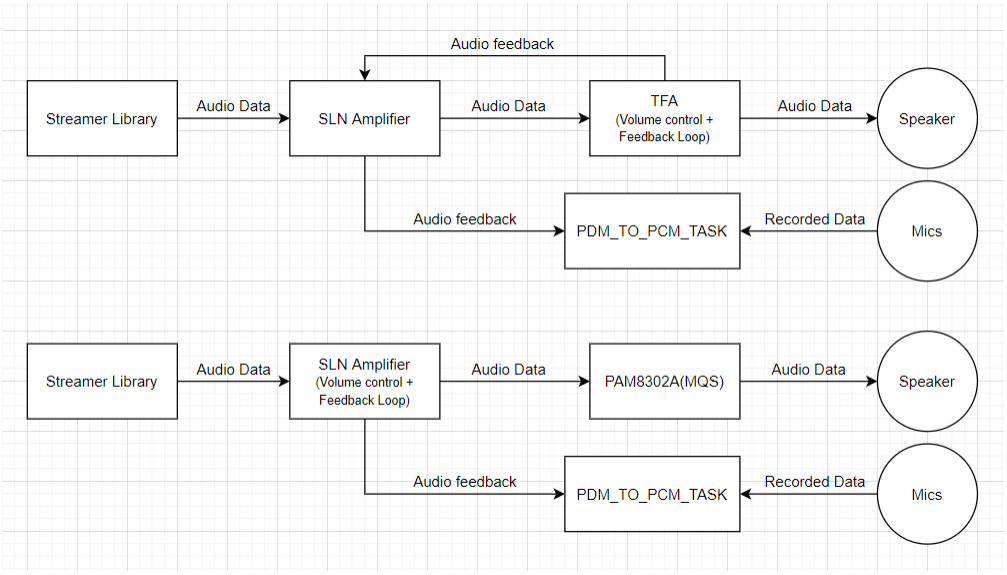

Figure 21. Audio data flow TFA vs MQS

For the speaker audio cancellation to operate properly, the feedback audio sent from the **sln_amplifier.c** must be aligned with the audio recorded by the microphones.

The alignment is done using a ringbuffer and adding a delay of size x at the beginning of the buffer before adding speaker data. The delay corresponds to two actual delays in code:

- **Constant:** Hardware dependent, calculated using sln_local2_rd_usb_aec_alignment_tool , and adjusted using the define **AMP_LOOPBACK_CONST_DELAY_US**.
- **Variable:** Calculated automatically using **AMP_LOOPBACK_GPT** (GPT2 Timer) and represents the difference between the current playback start time and the last microphone PING/PONG. This aligns the playback with the microphone input.

The audio alignment is done using the **sln_local2_rd_usb_aec_alignment_tool** found in the SDK package.

Finally, the **tfa_configurator** project is removed from the SDK because it is **TFA-**specific.

3.3 Other Software updates

The i.MX RT106S reference design supports two microphones instead of three microphones on the SLN-LOCAL2-IOT development kit, hence **pdm_pcm_definitions.h** was updated to reflect that.

The LED of the reference design hardware is controlled using **flexio**, and not using **pwm**, so a new LED driver was added: **sln_RT10xx_RGB_LED_driver_flexio**.c instead of **sln_RT10xx_RGB_LED_driver_pwm.c.** Therefore, **fsl_pwm** and **fsl_xbara** drivers were removed, while **fsl_flexio** driver was added.

4 Revision history

This table summarizes revisions to this document.

Table 6. Revision history

Revision number	Date	Substantive changes
0	19 April 2021	Initial release

5 References

1. NXP i.MX RT1060 Processor Reference Manual (IMXRT1060RM Reference Manual)
2. Cypress application note for optimized antenna design (AN91445 – Antenna Design and RF Layout Guidelines)
3. NXP application note for QSPI Boot Configuration (AN12108 - How to Enable Boot from QSPI Flash)
4. NXP application note for Slew Rate and Drive Strength Configuration (AN5078 - Influence of Pin Setting on System Function and Performance)
5. NXP application note for Spread Spectrum Configuration (AN12879 - How to Enable Spread Spectrum for RT Family)

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 19 April 2021

Document identifier: AN13208

arm