AN13606

Low-power Voice Wake-up and Recognition based on i.MX RT595

Rev. 0 — 01 April 2022 Application Note

Contents

1 Introduction

The i.MX RT595 is a dual-core microcontroller featuring an Arm Cortex-M33 core combined with a Cadence Tensilica Fusion F1 Audio DSP core. Both processors are running at frequencies of up to 200 MHz. It offers an audio subsystem supporting up to eight DMIC channels. The subsystem includes the dual-channel Digital PDM Microphone Interfacev(DMIC) and Hardware Voice Activity Detector (HWVAD).

The purpose of this document is to provide a method of voice wake-up and recognition using i.MX RT595. In this method, the application works in two states, listening state and processing state. This document introduces the operating mechanism, conversion conditions, and power consumption of each state. Both CM33 and DSP are involved in this application.

2 System architecture

In this document, the acquisition of voice data is done by a single-channel digital microphone. Figure 1 shows the timing diagram of the entire application. In combination with the HWVAD, it provides lowest power consumption in listening mode. Except for the short periods with DMA activities, the MCU can remain in Deep-sleep mode until the wave envelope detector of the HWVAD identifies an energy change event and issues an interrupt.

In fact, the system is in the listening state most of the time to save power consumption. It enters the processing state only when it detects that there is sound around or the voice band energy.

The following sections describe each state in blocks.

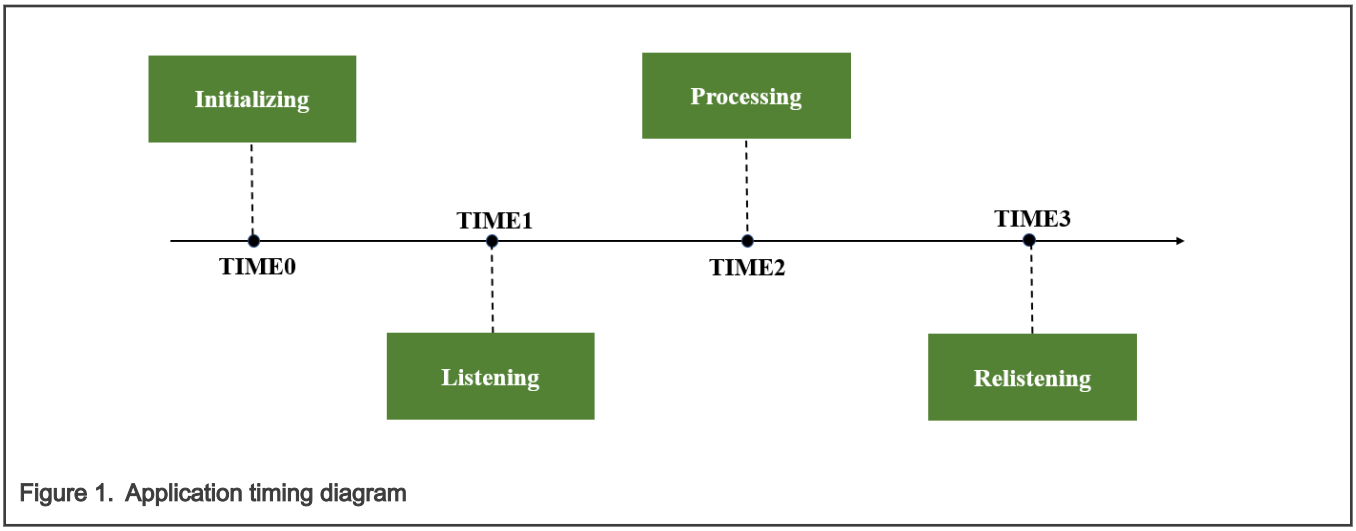

Figure 1. Application timing diagram

- **TIME0 -> Initializing**

  This process is mainly for the initialization of various peripherals, clocks, and memory. The specific work is as follows. In particular, to save power consumption, the DSP does not run.

1. Pin and clock initialization
2. DSP does NOT start
3. Init DMIC, DMA, HWVAD, LPOSC, No PLL
4. FRO trimmed at 96 MHz
5. CM33 = FRO/2 = 48 MHz, VDD Core = 0.75 V
6. SRAM 5 MB array and peripheral

- **TIME1 -> Listening**

  Listening mode follows initialization. In this state, the CM33 enters deep-sleep mode and DSP does not run. The DMIC clock source is set to LPOSC to ensure that all components in system are in low-power mode (both i.MXRT595 and DMIC).

  Set DMIC DMA FIFO depth to 16 and PCM sample rate to 16 kHz. Set register so that DMIC DMA can also work in deep-sleep mode. For details, see Work process.

  1. DMIC DMA wakes up every 1 ms use for DMIC data acquisition.
  2. CM33 wakes up every 10 ms to store audio history data.
  3. Keep storing latest data, discarding oldest data.

  When HWVAD detects a sound around or the voice band energy, the listening state is terminated and the system enters the next state.

- **TIME2 -> Processing**

  In the processing state, the CM33 is awakened and the DSP starts to run. The CM33 continues to control DMIC DMA to get data and store it in `ringbuf`. DSP reads data from `ringbuf` then transmits it to VIT for recognition.

  Semaphores2 (SEMA42) is used for SRAM access mutual exclusion.

  Check whether there is new sound every 3 seconds. If no, enter the next state.

- **TIME3 -> Re-enter Listening**

  Once no sound is detected within 3 seconds for a certain time, re-enter the listening state. The situation in this state is the same as **TIME1**.

3 Peripheral configuration

3.1 DMIC subsystem module

In DMIC audio acquisition, audio quality and power consumption are related to different PDM clock. Higher clock makes better audio quality and higher power consumption.

The DMIC subsystem includes digital PDM microphone interface and Hardware Voice Activity Detector (HWVAD). To balance power consumption and voice quality, two clock configurations are used for the subsystem. In addition, the function of HWVAD is to convert the application from the listening state to the processing state. After the conversion, the HWVAD ISR can be turned off and wait for the next listening state to arrive.

3.1.1 Clock configuration

3.1.1.1 PCM sample rate

The filter block for PDM to PCM conversion consists of four stages, as shown in Figure 2.

1. Cascaded-Integrator Comb (CIC) filter

   Convert the PDM stream from the digital microphone into PCM data with a given OverSampling Rate (OSR).

2. Decimate

Decimate by 2 and compensate for a roll-off at the upper limit of the audio band.

3. Decimate again

   Decimate the signal again by half, resulting in a PCM signal with the desired sample rate.

4. DC filter

   Remove any unwanted DC component in the audio signal.

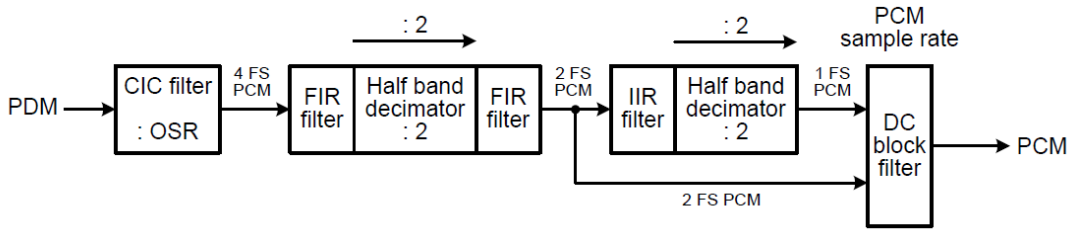

Figure 2. Principle structure of the PDM to PCM conversion

After the above four stages, the PCM sample rate can be obtained as follows.

$$PCM\ sample\ rate = \frac{DMIC\ clock\ rate}{2 \times OSR}, 2\ Fs\ used$$

Equation 1.

$$PCM\ sample\ rate = \frac{DMIC\ clock\ rate}{4 \times OSR}, 1\ Fs\ used$$

Equation 2.

3.1.1.2 Change clock during acquisition

To balance power consumption and voice quality, there are different clock configurations for listening state and processing state. This section discusses how to change the DMIC operating clock during acquisition. It is expected that PCM sample rate keeps same while PDM clock change.

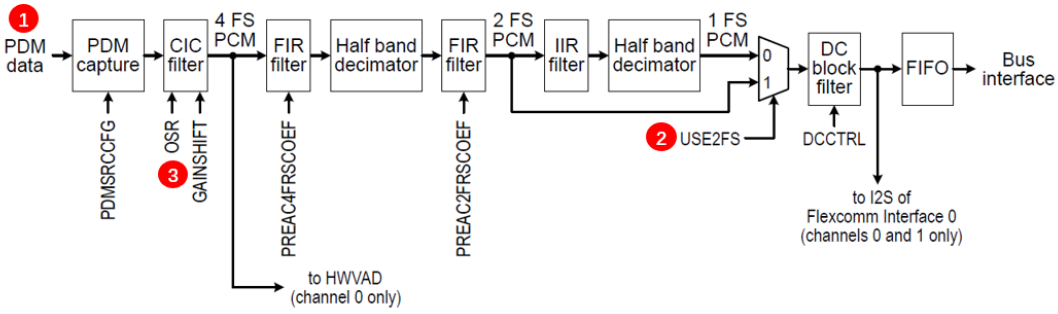

Figure 3. DMIC channel block diagram

There are three positions can be changed, as shown in red in Figure 3. To ensure that the PCM sample rate remains the same, the change must consider the formula in PCM sample rate.

- **Red mark 1**

  Change DMIC clock source and division factor, as shown in Figure 4. Different clock rate can be obtained at the red position 1 in Figure 3.

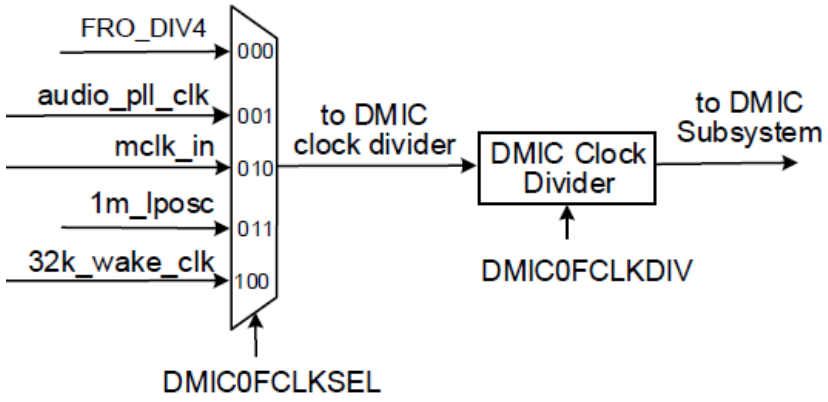

Figure 4. DMIC PDM clock selection

- **Red mark 2**

  If the USE2FS register bit is set to **1**, the IIR filter and half band decimator in Figure 3 can be ignored and the clock frequency does not need to be divided by 2 again. Calculate the PCM sampling rate according to Equation 1. Otherwise, use Equation 2.

- **Red mark 3**

  Position three is changing CIC decimator OSR.

  The OSR of the CIC decimator can be changed through the *DMIC_ConfigChannel* API. As the decimator OSR changes, the CIC integrator changes cumulatively and the gain shift must be changed accordingly.

  The CIC filter is full 32-bit dynamic range, and up to 24 bits can be selected for output. Once the order and OSR are determined, MAX CIC GAIN is fixed and it may overflow if it exceeds this value.

  i.MXRT600 PDM MEMS Microphone Audio Path Optimal Settings (document AN12590) suggests the gain setting in different OSR. The gain shift is designed to compensate for the Effective Number Of Bits (ENOB) reduction and to use a full 32-bit dynamic range on the CIC. The half band filters then produce 24-bit output. Theoretically, the gain shift value can be calculated using Equation 3.

$$gain\ shift\ =\ 32\ -\ ENOBENOB\ =\ ceiling\left(log_2(OSR^{Order})\right)\ +\ B\_Input,\ where\ B\_Input\ =\ 2$$

Equation 3.

  Among them, *B_Input* is PDM input bit depth.

  To be full scale, the recommended values of gain shift are as shown in Table 1.

Table 1. CIC setting

OSR	ORDER 5 ENOB	ORDER 5 Gain shift	/	ORDER 4 ENOB	ORDER 4 Gain shift
8	17	15	/	14	18
16	22	10	/	18	14
24	25	7	/	21	11
32	27	5	/	22	10
48	30	2	/	25	7

Table continues on the next page...

Table 1. CIC setting (continued)

OSR	ORDER 5 ENOB	ORDER 5 Gain shift	/	ORDER 4 ENOB	ORDER 4 Gain shift
64	32	0	/	26	6
96	35	N/A	/	29	3
128	37	N/A	/	30	2
192	40	N/A	/	33	N/A
Decimator switch automatically on ORDER 4 for all OSR > 64.					

According to Table 1, if OSR is changed along with the PDM clock, the gain shift must be changed at the same time.

When changing the PCM frequency by modifying the OSR, the decimator must be changed while resetting state, as shown in Code Listing 1.

```
DMIC_ResetChannelDecimator(DMIC0, (kDMIC_EnableChannel0 << i), true);
if((i % 2) == 0)
    DMIC_ConfigChannel(DMIC0, (dmic_channel_t) (kDMIC_Channel0 + i), kDMIC_Left,
&dmic_channel_cfg);
else
    DMIC_ConfigChannel(DMIC0, (dmic_channel_t) (kDMIC_Channel0 + i), kDMIC_Right,
&dmic_channel_cfg);
DMIC_ResetChannelDecimator(DMIC0, (kDMIC_EnableChannel0 << i), false);
```

Code Listing 1. Initialization and listening state DMIC clock setting

In addition, the output of voice data can be 16 bits or 24 bits. The SIGNEXTEND bit and SATURATEAT16BIT bit in DC Filter Control register change with the choice of the number of bits.

- When output 16 bits, saturation = 1, SignExtend = 0
- When output 24 bits, saturation = 0, SignExtend = 1

3.1.1.3 Actual clock configuration

- Initialization and listening state

  When the application is in the initialization and listening state, LPOSC is used as the DMIC clock source. It is low power and the audio quality is less important in listening mode.

  To ensure that the PCM sampling rate close to 16 k, the parameters are configured as follows.

  - Clock source: LPOSC
  - division factor: 2
  - OSR: 16
  - Gainshift: 8
  - Saturation = 1
  - SignExtend = 0
  - Use2Fs = 1

NOTE

$$PCM\ sample\ rate = \frac{DMIC\ clock\ rate}{2 \times OSR} = \frac{1000000/2}{2 \times 16} = 15625\text{Hz}$$

Equation 4.

Code Listing 2 shows the corresponding configuration code.

```
static dmic_channel_config_t dmic_channel_cfg;
CLOCK_AttachClk(kLPOSC_to_DMIC);
CLOCK_SetClkDiv(kCLOCK_DivDmicClk, 1);
memset(&dmic_channel_cfg,0U,sizeof(dmic_channel_config_t));
dmic_channel_cfg.divhfclk   = kDMIC_PdmDiv1;
dmic_channel_cfg.osr          = 16;
dmic_channel_cfg.gainshft   = 8U;
dmic_channel_cfg.preac2coef    = kDMIC_CompValueZero;
dmic_channel_cfg.preac4coef    = kDMIC_CompValueZero;
dmic_channel_cfg.dc_cut_level   = kDMIC_DcCut155;
dmic_channel_cfg.post_dc_gain_reduce = 0U;
dmic_channel_cfg.saturate16bit  = 1U;
dmic_channel_cfg.sample_rate   = kDMIC_PhyFullSpeed;
#if defined(FSL_FEATURE_DMIC_CHANNEL_HAS_SIGNEXTEND) &&
(FSL_FEATURE_DMIC_CHANNEL_HAS_SIGNEXTEND)
    dmic_channel_cfg.enableSignExtend = false;
#endif

DMIC_Use2fs(DMIC0, true);
DMIC_ConfigChannel(DMIC0, kDMIC_Channel0, kDMIC_Left, &dmic_channel_cfg);
```

Code Listing 2. Initialization and listening state DMIC clock setting

In addition, we can have DMA service during deep-sleep mode without waking up Arm processor by using HWWAKE register. Peripheral FIFO levels trigger these wake-ups. When DMA transfer completes, DMA interrupt awakes Arm.

When a peripheral that is able to operate (at least in some modes) during deep-sleep reaches its programmed FIFO threshold, it can cause bus clocks to be temporarily enabled. During that time, DMA can move data from the peripheral to memory or from memory to the peripheral. When DMA completes, full deep-sleep mode is resumed.

To achieve this purpose and enable HWVAD to wake up the device from deep-sleep mode, perform the settings as shown in Code Listing 3.

```
SYSCTL0->HWWAKE |= SYSCTL0_HWWAKE_DMICWAKE_MASK | SYSCTL0_HWWAKE_DMAC0WAKE_MASK;
EnableDeepSleepIRQ(HWVAD0_IRQn);
```

Code Listing 3. HWWAKE setting

In fact, we tried to set the CM33 clock to 12 M or 48 M respectively in the listening state. The results show that when the CM33 is running at 12 M, the IDDCORE is about 200 μA, and when it is running at 48 M, the IDDCORE is about 180 μA.

As mentioned above, CM33 is briefly awakened and does something in the listening state. When the frequency of CM33 is lower, it takes a longer time to enter the deep-sleep mode again and has higher power consumption in this state.

- Processing state

  When the application is in the processing state, FRO is used as the DMIC clock source. Since FRO has been trimmed and enabled as 96 M, FRO_DIV4 clock is 24 M.

  The parameter setting must meet two conditions:

  1. The sampling rate is also around 16 k.

2. The voice amplitude cannot be too different from the previous one.

From the CIC setting in Table 1, when the OSR changes, the gain shift also changes. Also, when the OSR is 16 and 25 respectively, the gain shift difference is about 3.

- Clock source: FRO_DIV4
- division factor: 30
- OSR: 25
- Gainshift: 5
- Saturation = 1
- SignExtend = 0
- Use2Fs = 1

$$PCM\ sample\ rate = \frac{DMIC\ clock\ rate}{2 \times OSR} = \frac{24000000/30}{2 \times 25} = 16000\text{Hz}$$

Equation 5.

Code Listing 4 shows the corresponding configuration code.

```
CLOCK_AttachClk(kFRO_DIV4_to_DMIC);
CLOCK_SetClkDiv(kCLOCK_DivDmicClk, 60);
DMIC_ResetChannelDecimator(DMIC0, kDMIC_EnableChannel0, true);
dmic_channel_cfg.osr = 25;
dmic_channel_cfg.gainshft = 5;
DMIC_ConfigChannel(DMIC0, kDMIC_Channel0, kDMIC_Left, &dmic_channel_cfg);
DMIC_ResetChannelDecimator(DMIC0, kDMIC_EnableChannel0, false);
```

Code Listing 4. Processing state DMIC clock setting

3.1.2 HWVAD

HWVAD is used to determine whether a sound is detected and it implements a wave envelope detector and a floor noise envelope detector. The interrupt is issued when a specific delta between the signal and the noise result is detected. The HWVAD can be active when the DMIC interface is active.

For HWVAD, its input signal can only come from DMIC channel 0 and it is optimized for PCM signals with 16 kHz sampling frequency.

Figure 5 shows the HWVAD interrupt structure.

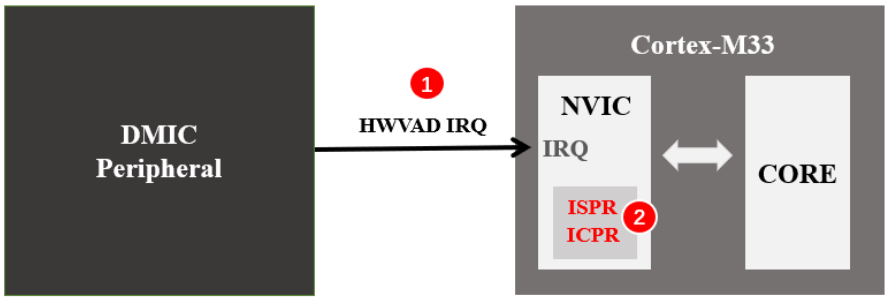

Figure 5. HWVAD interrupt structure

Below describes the details for the **red marks** in Figure 5:

- **Red mark 1**

  It enables the HWVAD interrupt in the NVIC.

- **Red mark 2**

  It indicates the interrupt-related registers in the NVIC, Interrupt Set Pending Registers (ISPR), and Interrupt Clear Pending bits (ICPR).

In this document, whether the application detects the sound as the basis for state conversion.

In listening state, CM33 is in deep-sleep mode and DSP does not run. Enable the HWAVD interrupt in NVIC and interrupt wake-up in the STARTEN1 register. When the sound is detected, the DSP starts/resumes operation, CTimer starts timing, and the HWVAD interrupt does not need to be enabled in the NVIC (turning off the red mark one in Figure 5). Afterwards clear interrupt request as shown in Figure 6.

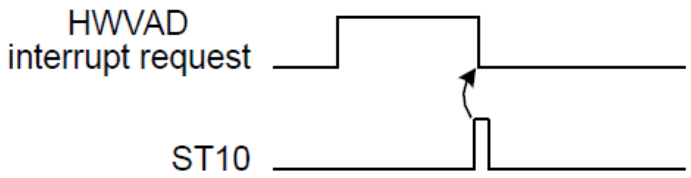

Figure 6. Reset HWVAD interrupt flag

Code Listing 5 shows the process.

```
void (void)
{
  volatile int i;
  /* reset hwvad internal interrupt */
  DMIC_CtrlClrIntrHwvad(DMIC0, true);
  /* wait for HWVAD to settle */
  if(dsp_started == 0) {

      dsp_started = 1;
  //     DSP_Start();
      DSP_RESUME();
      DisableDeepSleepIRQ(HWVAD0_IRQn);
      CTIMER_StartTimer(CTIMER2);
  }

  for (i = 0; i <= 500U; i++)
  {
  }
  /*HWVAD Normal operation */
  DMIC_CtrlClrIntrHwvad(DMIC0, false);
}
```

Code Listing 5. HWVAD interrupt callback

In processing state, disables the HWVAD interrupt in NVIC and interrupt wake-up in the STARTEN1 register, so it will not enter the `DMIC0_HWVAD_Callback` function. The CTimer interrupt time is set to 3 s. Whenever an interrupt comes, read the ISPR bit in the red mark 2 (Figure 5) to determine whether there is a sound. If a sound has been detected, clear the pending bit through ICPR

and HWVAD interrupt request without any other processing; otherwise, enable the HWVAD interrupt and enter the listening state at the same time.

Code Listing 6 shows the process.

```
void ctimer_match0_callback(uint32_t flags)
{
   uint32_t hwvad_flag;
   hwvad_flag = NVIC_GetPendingIRQ(HWVAD0_IRQn);
   if(hwvad_flag)
   {
        DMIC_CtrlClrIntrHwvad(DMIC0, true);
        /* Delay to clear first spurious interrupt and let the filter converge */
        for (int i = 0; i <= 500U; i++)
        {
        }
         /*HWVAD Normal operation */
         DMIC_CtrlClrIntrHwvad(DMIC0, false);
         NVIC_ClearPendingIRQ(HWVAD0_IRQn);
     }
     else
     {
         dsp_started = 0;
         EnableDeepSleepIRQ(HWVAD0_IRQn);
      //    DSP_Stop();
         DSP_POWERDOWN();
         CTIMER_StopTimer(CTIMER2);
      }
}
```

Code Listing 6. Ctimer interrupt callback

3.2 Semaphores2

Semaphpres2 (SEMA42) is a memory-mapped module that provides robust hardware support needed in multicore systems for implementing semaphores and provides a simple mechanism to achieve **lock and unlock** operations via a single - write access. The hardware semaphore module provides hardware-enforced gates and other useful system functions related to the gating mechanisms.

In this document, SMEA42 is used for mutual exclusion between CM33 and DSP to access shared memory. Before each master accesses the shared memory, it locks the memory through SMEA42 and unlocks it after the access is complete. For details, see **Chapter 43** in UM.

Code Listing 7 shows how to lock and unlock.

```
void Core_Lock(uint8_t gate)
{
#if defined(CPU_MIMXRT595SFFOC_cm33)
    SEMA42_Lock(SEMA42, gate, 1);
#elif defined(CPU_MIMXRT595SFFOC_dsp)
    SEMA42_Lock(SEMA42, gate, 3);
#endif
}

void Core_unLock(uint8_t gate)
{
```

```
    SEMA42_Unlock(SEMA42, gate);
}
```

Code Listing 7. Example 7 Method of locking and unlocking

4 Work process

This chapter systematically describes the voice wake-up and recognition work process.

- Current work strategy describes the current work strategy, including the mutual conversion conditions between listening state and processing state.
- Other strategies describes other strategies for the transition from the processing state to the listening state.

4.1 Current work strategy

Figure 7 shows the transition from sleep state to processing state. The areas marked with red numbers must be paid attention to.

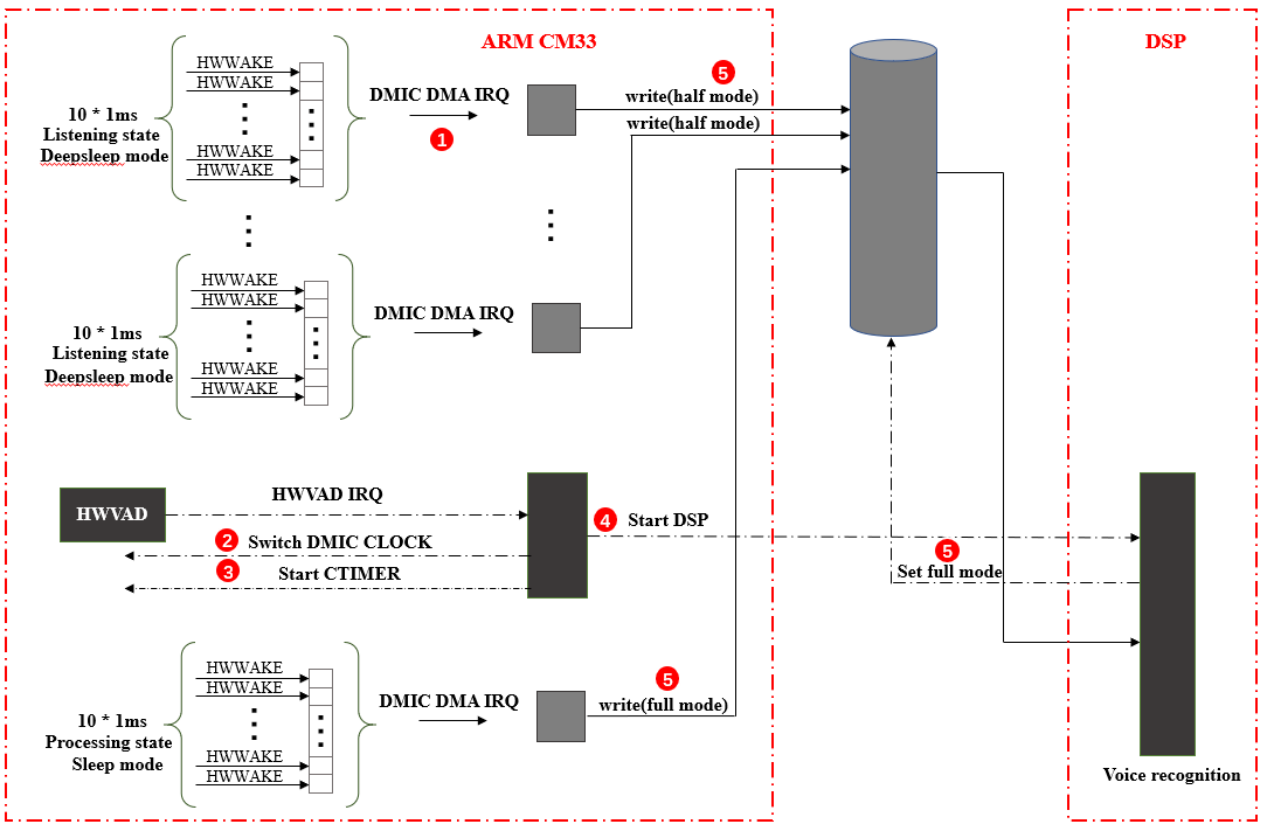

Figure 7. Voice wake-up and recognition diagram

Below describes the details for the red marks in Figure 7:

- **Red mark 1: DMIC DMA IRQ**

  In clock distribution, the PCM sampling rate is 16 kHz, that is, there are 16 samples in 1 ms. The DMIC FIFO trigger level is set to 16 entries, which is equivalent to HWWAKE coming every 1 ms. DMA transfer is set to 10 ms data size, which is equivalent to DMA IRQ coming every 10 ms. The setting of the HWWAKE register in Code Listing 2 causes both of them to wake up the CM33 briefly.

The PMIC_MODESEL0 and PMIC_MODESEL1 pins determine the CM33 status at this time, as shown in Figure 8. Determine the CM33 status at this time, where **00** stands for active/sleep mode, **10** stands for deep-sleep mode.

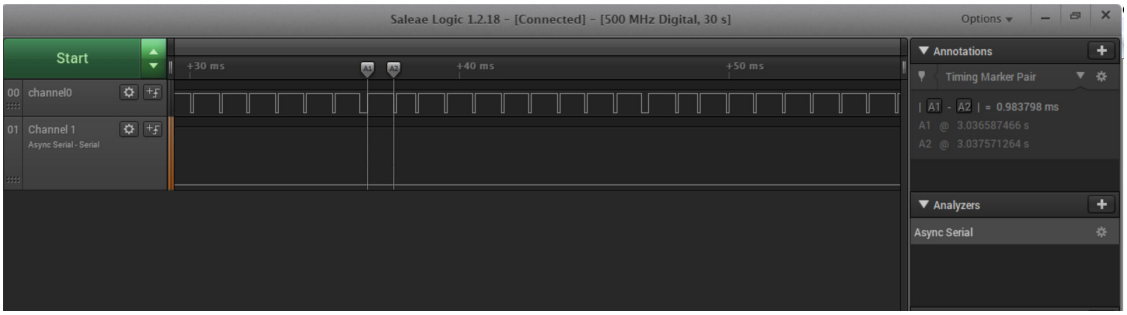

Figure 8. PMIC_MODESEL state diagram

- **Red mark 2: Switch DMIC Clock**

  For details, see Change clock during acquisition.

- **Red mark 3: Start CTIMER**

  Determine whether a sound is detected within 3 seconds in each CTimer interrupt, and whether to re-enter the listening state according to the result. For details, see HWVAD.

- **Red mark 4: Start DSP**

  There are two ways to stop DSP operation: DSP Stall and DSP Power down. The difference between them is that DSP operation can be resumed by exiting **DSP Stall** mode without DSP reset. But if DSP is power down, DSP must be reset for another run. For details about the two stop methods and the corresponding start methods, see *Using DSP in Lowpower Design*.

  If the XOS system is not used in the DSP application, use the DSP Stall method to stop or run the DSP (that is, call *DSP_Stop* and *DSP_Start* API). Otherwise, use another method.

- **Red mark 5: Half mode/Normal mode**

  To ensure the integrity of the data received by the DSP, it is necessary to keep some historical data in the `ringbuf`.

  In the listening state, CM33 uses Half mode to store DMIC DMA data in `ringbuf`. By saving the latest data and discarding the oldest data, the stored historical data does not exceed half the size of the entire `ringbuf`.

  When entering the processing state, the DSP starts to run and sets the access `ringbuf` mode to Normal mode. The DSP side reads data from the `ringbuf` and recognizes it through VIT. The CM33 side continues to write data to the `ringbuf` using the Normal mode.

4.2 Other strategies

In fact, the specific voice recognition algorithm determines when to transition from the processing state to the listening state. In this document, the algorithm is VIT and the condition for the transition is no sound is detected within 3 seconds. That is, the pending bit in the ISPR register is not set in the CTimer interrupt.

For VIT algorithm, the voice command must be spoken within 1.5 seconds after the wake-up word is recognized. Otherwise, the wake-up word must be spoken again. Ideally, to reduce the time in the processing state, we can choose the other strategy. When wake-up word is detected, reset the CTimer counter.

5 Power measurement

There are various power domains in i.MX RT595 EVK. The power domains are supplied by different power rails, as illustrated in *i.MX RT500 Power Management* (document AN13162).

On i.MX RT595 EVK, jumpers are reserved for power measurement in different domains. Digital multimeter is placed in series instead of jumpers.

Among them, the power domains that this document focuses on are summarized in Table 2.

Table 2. i.MX RT595 EVK power domains

Power domain	Power supply	Power sink	Jumper
VDD core	PMIC_SW1_OUT	CM33, DSP, SRAM, Peripherals IP	JS25
VDD1V8	PMIC_SW2_OUT	Analog (PLL, OSC, ADC), OTP, PMC	JS30
VDDIO0	MCU_1V8 (PMIC_SW2_OUT)	FlexSPI0 NOR FLASH IOs	JS20
VDDIO1	MCU_1V8 (PMIC_SW2_OUT)	DMIC IOs	JS21

5.1 Power measurement in listening state

In this state, the CM33 enters deep-sleep mode and DSP does not run. DMIC DMA wakes up every 1 ms use for DMIC data acquisition, CM33 wakes up every 10 ms to store voice history data.

As mentioned above, the CM33 is briefly awakened and does something in the listening state. When the frequency of CM33 is lower, it takes a longer time to enter the deep-sleep mode again, so it has higher power consumption in listening state but lower power consumption in processing state. It is a trade-off for power consumption in diff state.

Therefore, we list the power consumption comparison of the CM33 running at 48 M and 12 M.

When running at 48 M, the `vddcore` voltage is around 0.621 V most of the time, and at 0.75 V when it is briefly awakened. The average voltage is 0.668 V. Table 3 shows the power measurement result in the listening state at 48 M frequency.

When running at 12 M, Table 4 shows the result at 12 M frequency.

Table 3. Power measurement result under 48 M

Power domain	Jumper	Current
VDD core	JS25	170 µA
VDD1V8	JS30	0.21 mA
VDDIO0_1V8	JS20	110.5 µA
VDDIO1_1V8	JS21	58.4 µA

Table 4. Power measurement result under 12 M

Power domain	Jumper	Current
VDD core	JS25	210 µA
VDD1V8	JS30	0.21 mA
VDDIO0_1V8	JS20	110.5 µA
VDDIO1_1V8	JS21	58.4 µA

5.2 Power measurement in processing state

In this state, the CM33 side is still storing the DMIC data in the `ringbuf`. The DSP side is getting the data from the `ringbuf` and performing VIT identification. For the VIT algorithm, the DSP must run at least 96 M.

Table 5 shows the power measurement result in processing state when CM33 runs at 48 M and DSP runs 96 M. Table 6 shows the result when CM33 runs at 12 M and DSP runs at 96 M.

Table 5. Power measurement result @ CM33 48 M DSP 96 M

Power domain	Jumper	Current
VDD core	JS25	7.39 mA
VDD1V8	JS30	0.42 mA
VDDIO0_1V8	JS20	115.4 µA
VDDIO1_1V8	JS21	95.8 µA

Table 6. Power measurement result @CM33 12 M DSP 96 M

Power domain	Jumper	Current
VDD core	JS25	6.13 mA
VDD1V8	JS30	0.42 mA
VDDIO0_1V8	JS20	115.1 µA
VDDIO1_1V8	JS21	95.8 µA

6 Conclusion

This document describes a method of voice wake-up and recognition using i.MX RT595. To accomplish this mothod, the following aspects must be considered:

1. There are two mainly states in the application, listening and processing. Considering the work content of each state and the transition conditions between states.
2. DSP access to `ringbuf` must retain some historical data, especially when switching from listening mode to processing mode.
3. DMIC several clock conversion methods.
4. Two DSP operation and stop modes and their respective applicable conditions.

7 References

1. *i.MX RT500 Low-Power Crossover MCU Reference Manual with Addendum* (document IMXRT500RM)
2. MIMXRT595-EVK Schematic diagram
3. *i.MXRT600 PDM MEMS Microphone Audio Path Optimal Settings* (document AN12590)
4. *i.MX RT500 Power Management* (document AN13162)
5. *Using DSP in Lowpower Design*

8 Revision history

Rev.	Date	Description
0	1 April 2022	Initial release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Airfast — is a trademark of NXP B.V.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

Cadence — the Cadence logo, and the other Cadence marks found at www.cadence.com/go/trademarks are trademarks or registered trademarks of Cadence Design Systems, Inc. All rights reserved worldwide.

CodeWarrior — is a trademark of NXP B.V.

ColdFire — is a trademark of NXP B.V.

ColdFire+ — is a trademark of NXP B.V.

EdgeLock — is a trademark of NXP B.V.

EdgeScale — is a trademark of NXP B.V.

EdgeVerse — is a trademark of NXP B.V.

eIQ — is a trademark of NXP B.V.

FeliCa — is a trademark of Sony Corporation.

Freescale — is a trademark of NXP B.V.

HITAG — is a trademark of NXP B.V.

ICODE and I-CODE — are trademarks of NXP B.V.

Immersiv3D — is a trademark of NXP B.V.

I2C-bus — logo is a trademark of NXP B.V.

Kinetis — is a trademark of NXP B.V.

Layerscape — is a trademark of NXP B.V.

Mantis — is a trademark of NXP B.V.

MIFARE — is a trademark of NXP B.V.

MOBILEGT — is a trademark of NXP B.V.

NTAG — is a trademark of NXP B.V.

Processor Expert — is a trademark of NXP B.V.

QorIQ — is a trademark of NXP B.V.

SafeAssure — is a trademark of NXP B.V.

SafeAssure — logo is a trademark of NXP B.V.

StarCore — is a trademark of NXP B.V.

Synopsys — Portions Copyright © 2021 Synopsys, Inc. Used with permission. All rights reserved.

Tower — is a trademark of NXP B.V.

UCODE — is a trademark of NXP B.V.

VortiQa — is a trademark of NXP B.V.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 01 April 2022
Document identifier: AN13606