GigaDevice Semiconductor Inc.

ARM® Cortex®-M3/4/23/33 32-bit MCU

Application Note
AN037

Table of Contents

List of Figures

List of Table

1. Introduction

A high performance digital filter module (HPDF) for external sigma delta (Σ-Δ) modulator is integrated in GD32W51x. HPDF supports SPI interface and Manchester-coded single-wire interface. The external sigma delta modulator can be connected with MCU by the serial interface, and the serial data stream output by sigma delta modulator can be filtered. In addition, HPDF also supports the parallel data stream input function to filter the data in the internal memory of the MCU.

This article provides a solution for real-time playback of audio signals based on HPDF and I2S modules, and details the configuration method of the HPDF module.So that customers can learn to use the HPDF module of GD32 MCU more quickly.

1.1. HPDF Characteristics

- Two multiplex digital serial input channels
  - configurable SPI and Manchester interfaces;
- Two internal digital parallel input channels
  - input with up to 16-bit resolution;
- Configurable Sinc filter and integrator
  - the order and oversampling rate (decimation rate) of Sinc filter can be configured;
  - sampling rate of configurable integrator;
- Threshold monitor function
  - independent Sinc filter, configurable order and oversampling rate (decimation rate);
  - configurable data input source: serial channel input data or HPDF output data;
- Malfunction monitor function
  - A counter with 8 bits is used to monitor the continuous 0 or 1 in the serial channel input data stream;
- Extreme monitor function
  - store minimum and maximum values of output data values of HPDF;
- Up to 24-bit output data resolution
- Clock signal can be provided to external sigma delta modulator
  - provide configurable clock signal by the CKOUT pin;
- Flexible conversion configuration function
  - the conversion channel is divided into regular group and inserted group;
  - support multiple conversion modes and startup modes;
- HPDF output data is in signed format

1.2. HPDF and Σ-Δ modulator

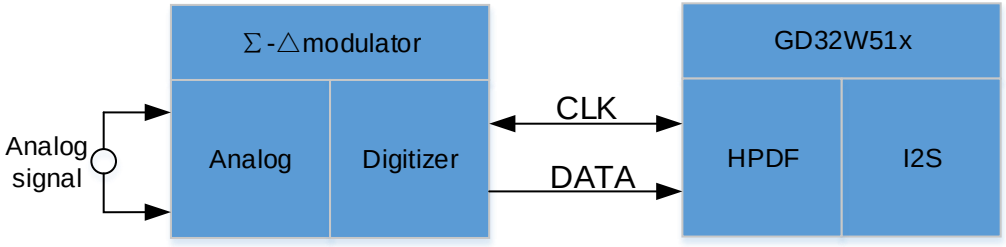
The HPDF module needs to be used in conjunction with an external Σ-Δ modulator. The connection method is shown in ***Figure 1-1. HPDF and Σ-Δ modulator connection block***

diagram.

The external sigma-delta modulator converts the analog signal into a digital 1-bit data stream (DATA and CLK signals) to process the external analog signal. 1-bit data stream is a fast serial line stream of logic 1 and 0 which DATA signal is sampled by CLK (clock signal). The average value of these logic 1s and 0s calculated over a sufficiently long duration represents the analog input value. The duration of the averaging period determines the accuracy of the analog input signal capture.

The GD32 microcontroller HPDF peripheral takes the average of the 1-bit stream, and the HPDF acquires and processes the 1-bit data stream (digital filtering, averaging). HPDF outputs data samples at a higher resolution than the data rate of the input 1-bit stream. The HPDF digital filter sets the output resolution and data rate.

Figure 1-1. HPDF and Σ-Δ modulator connection block diagram



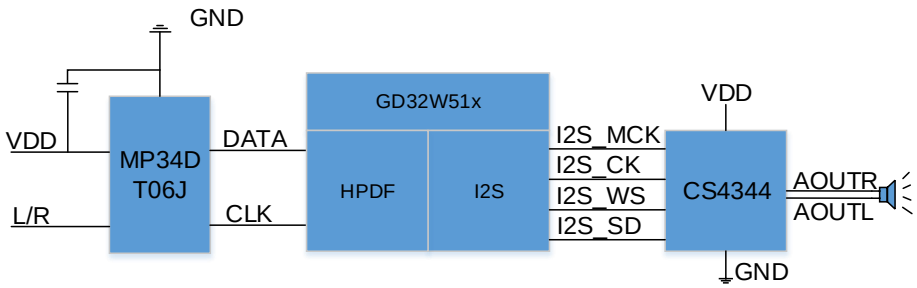
As shown in ***Figure 1-1. HPDF and Σ-Δ modulator connection block diagram***, the HPDF module of GD32 MCU uses two wires to connect with the Σ-Δ modulator as the signal wire for serial communication. The two signal lines are respectively a data (DATA) line and a clock (CLK) line. The clock signal can be generated by the sigma-delta modulator or by the HPDF module.

2. MAC audio playb ack solution

2.1. Mono solution

The mono audio playback scheme based on the HPDF module is shown in ***Figure 2-1. HPDF mono audio solution***. In this scheme, the HPDF module uses a serial channel to connect with an external sigma-delta modulator. Since there is only one sigma-delta modulator connected externally, this solution can only collect left/right channel data.

Figure 2-1. HPDF mono audio solution

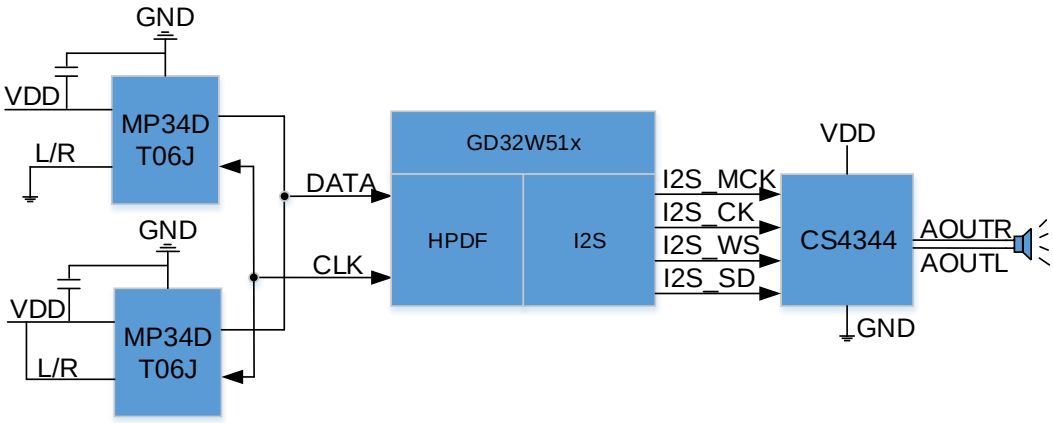


2.2. Stereo solution

The stereo audio playback scheme based on the HPDF module is shown in ***Figure 2-2. HPDF stereo audio solution***. In this scheme, the HPDF module uses two serial channels to connect with two external Σ-Δ modulators, and the signal line shares a set of DATA and CLK signal lines.

The channel pin redirection function of HPDF needs to be used in the stereo solution. Channel pin redirection means that the pins of serial channel 0 of the HPDF module can be configured as pins of channel 1, that is, channel 0 can read information from the DATIN1 and CKIN1 pins of the MCU. The pin redirection function is suitable for collecting audio data from PDM microphones. The PDM microphone audio signal contains data and clock signals. The data is divided into left/right channel data. The left channel data is collected on the rising edge of the clock signal, and the right channel data is collected on the falling edge of the clock signal.

Figure 2-2. HPDF stereo audio solution



3. The configuration of HPDF in the stereo solution

3.1. The process of converting the PDM data stream

When PDM microphone data stream is input into serial channel, its configuration process is as follows:

1. Select the HPDF serial channel 1 of PDM microphone data stream input.
2. Write 0 to CHPINSEL bit of channel 1 in HPDF_CHxCTL register, and input pin of channel 1 is own pin, DATAINx and CKINx. When SITYP[1:0] = 2b'00, the serial data stream is sampled at the rising edge of the clock signal, that is, the input of channel 1 is the left channel data.
3. Set the CHPINSEL to 1 in channel 0, and the DATAINx and CKINx pins will be used for channel 0. When SITYP[1:0] = 2b'01, the serial data stream is sampled at the falling edge of the clock signal, that is, the input of channel 0 is the right channel data.
4. Configure channel 0 and channel 1 with corresponding filters to filter the left and right channel data of PDM microphone.

The CKIN pin in the above process can also be changed to use the CKOUT pin shared by HPDF.

3.2. Clock and audio sampling rate

To output audio data with a specified sampling frequency, HPDF needs to configure the output clock, channel filter parameters, and integrator parameters of the HPDF module. Using serial data stream as input source, the calculation formula of HPDF output audio data is shown in ***Table 3-1. Audio data output rate***.

Table 3-1. Audio data output rate

Input source	Conversion mode	Filter type	Maximum output data rate (samples/second)
Serial input	Non-fast mode (FAST=0)	$Sinc^X$	$\frac{f_{CKOUT}}{SFOR\times(IOR-1+SFO)+(SFO+1)}$
	Non-fast mode (FAST=0)	FastSinc	$\frac{f_{CKOUT}}{SFOR\times(IOR-1+4)+(2+1)}$
	Fast mode (FAST=1)	FastSinc and $Sinc^X$	$\frac{f_{CKOUT}}{SFOR\times IOR}$

Note: In the table, SFOR means filter oversampling rate, SFO means filter order, IOR means integrator oversampling rate, f_{CKOUT} means HPDF output clock.

3.3. HPDF module configuration

Taking the fast mode as an example, suppose the audio sampling rate to be configured is 16KHz, the filter of the HPDF module selects a third-order Sinc filter, the oversampling rate is 64, and the integrator is bypassed (IOR = 1), then f_{CKOUT} = 1.2042MHz. If HPDF selects CK_HPDF as the clock source, else CK_HPDF = CK_APB2 = 90MHz, and the clock division factor of CKOUT is 88.

Refer to the HPDF user manual, it can be seen that the Sinc3 filter, when SFOR = 64, the maximum effective resolution of the HPDF output data is 19 bits, and the output audio data does not need to be shifted right. The detailed configuration of the HPDF module is shown in ***Table 3-2. Configuration of HPDF in GD project***.

Table 3-2. Configuration of HPDF in GD project

```
/*!
    \brief        configure the HPDF
    \param[in]    none
    \param[out] none
    \retval       none
*/
void hpdf_config(void)
{
    hpdf_channel_parameter_struct hpdf_channel_init_struct;
    hpdf_filter_parameter_struct hpdf_filter_init_struct;
    hpdf_rc_parameter_struct hpdf_rc_init_struct;

    /* reset HPDF */
    hpdf_deinit();

    /*   initialize the parameters */
    hpdf_channel_struct_para_init(&hpdf_channel_init_struct);
    hpdf_filter_struct_para_init(&hpdf_filter_init_struct);
    hpdf_rc_struct_para_init(&hpdf_rc_init_struct);

    /* configure serial clock output */
    hpdf_clock_output_config(SERIAL_SYSTEM_CLK,CKOUTDIV_88,CKOUTDM_ENABLE);

    /* initialize HPDF channel0 */
    hpdf_channel_init_struct.data_packing_mode       = DPM_STANDARD_MODE;
    hpdf_channel_init_struct.channel_pin_select      = CHPINSEL_NEXT;
    hpdf_channel_init_struct.ck_loss_detector        = CLK_LOSS_DISABLE;
    hpdf_channel_init_struct.malfunction_monitor     = MM_ENABLE;
    hpdf_channel_init_struct.spi_ck_source           = INTERNAL_CKOUT;
    hpdf_channel_init_struct.channel_multiplexer     = SERIAL_INPUT;
```

```
hpdf_channel_init_struct.serial_interface          = SPI_FALLING_EDGE;
hpdf_channel_init_struct.calibration_offset        = 0;
hpdf_channel_init_struct.right_bit_shift           = 0;
hpdf_channel_init_struct.tm_filter                 = TM_FASTSINC;
hpdf_channel_init_struct.tm_filter_oversample      = TM_FLT_BYPASS;
hpdf_channel_init_struct.mm_break_signal           = DISABLE;
hpdf_channel_init_struct.mm_counter_threshold      = 255;
hpdf_channel_init_struct.plsk_value                = 0;
hpdf_channel_init(CHANNEL0, &hpdf_channel_init_struct);

/* initialize HPDF channel1 */
hpdf_channel_init_struct.channel_pin_select        = CHPINSEL_CURRENT;
hpdf_channel_init_struct.serial_interface          = SPI_RISING_EDGE;
hpdf_channel_init(CHANNEL1, &hpdf_channel_init_struct);

/* initialize HPDF filter0 and filter1 */
hpdf_filter_init_struct.sinc_filter                = FLT_SINC3;
hpdf_filter_init_struct.sinc_oversample            = FLT_OVER_SAMPLE_64;
hpdf_filter_init_struct.integrator_oversample      = INTEGRATOR_BYPASS;
hpdf_filter_init(FLT0, &hpdf_filter_init_struct);
hpdf_filter_init(FLT1, &hpdf_filter_init_struct);

/* initialize HPDF filter0 regular conversions */
hpdf_rc_init_struct.fast_mode          = FAST_ENABLE;
hpdf_rc_init_struct.rcs_channel        = RCS_CHANNEL0;
hpdf_rc_init_struct.rcdmaen            = RCDMAEN_ENABLE;
hpdf_rc_init_struct.continuous_mode = RCCM_ENABLE;
hpdf_rc_init(FLT0, &hpdf_rc_init_struct);

/* initialize HPDF filter1 regular conversions */
hpdf_rc_init_struct.rcs_channel        = RCS_CHANNEL1;
hpdf_rc_init(FLT1, &hpdf_rc_init_struct);

/* enable channel */
hpdf_channel_enable(CHANNEL0);
hpdf_channel_enable(CHANNEL1);
/* enable filter */
hpdf_filter_enable(FLT0);
hpdf_filter_enable(FLT1);
/* enable the HPDF module globally */
hpdf_enable();
/* enable regular channel conversion by software */
hpdf_rc_start_by_software(FLT0);
```

```
    hpdf_rc_start_by_software(FLT1);
}
```

As shown in *Table 3-2. Configuration of HPDF in GD project*, HPDF uses two channels as regular group conversion channels. Channel 0 samples the right channel data on the rising edge of the CKOUT clock, and channel 1 samples the left channel data on the falling edge.

3.4. I2S and DMA module configuration

The audio data converted by HPDF is moved to buffer through DMA. Due to the stereo solution, two DMA channels are required. Place the data in the left and right channel data buffer respectively, and then alternately store the left and right channel data in the audio playback data buffer in turn. I2S reads audio data from the audio playback data Buffer. The DMA configuration is shown in *Table 3-3. Configuration of DMA in GD project*.

Table 3-3. Configuration of DMA in GD project

```
/*!
    \brief      configure DMA
    \param[in]  none
    \param[out] none
    \retval     none
*/
void dma_config(void)
{
    dma_single_data_parameter_struct dma_init_parameter;
    dma_single_data_para_struct_init(&dma_init_parameter);
    /* deinitialize DMA1_CH1 */
    dma_deinit(DMA1, DMA_CH1);
    /* configure DMA1_CH1 */
    dma_init_parameter.periph_addr          = (int32_t)&HPDF_FLTyRDATA(FLT0);
    dma_init_parameter.periph_inc           = DMA_PERIPH_INCREASE_DISABLE;
    dma_init_parameter.memory0_addr         = (uint32_t)pcm_right_data;
    dma_init_parameter.memory_inc = DMA_MEMORY_INCREASE_ENABLE;
    dma_init_parameter.periph_memory_width = DMA_PERIPH_WIDTH_32BIT;
    dma_init_parameter.circular_mode        = DMA_CIRCULAR_MODE_ENABLE;
    dma_init_parameter.direction            = DMA_PERIPH_TO_MEMORY;
    dma_init_parameter.number               = BUFFER_SIZE;
    dma_init_parameter.priority             = DMA_PRIORITY_ULTRA_HIGH;
    dma_single_data_mode_init(DMA1, DMA_CH1, &dma_init_parameter);
    /* connect DMA1_CH1 to HPDF_FLT0 */
    dma_channel_subperipheral_select(DMA1, DMA_CH1, DMA_SUBPERI7);
    /* enable DMA channel */
    dma_channel_enable(DMA1, DMA_CH1);
```

```
    /* deinitialize DMA1_CH2 */
    dma_deinit(DMA1, DMA_CH2);
    /* configure DMA1_CH2 */
    dma_init_parameter.periph_addr   = (int32_t)&HPDF_FLTyRDATA(FLT1);
    dma_init_parameter.memory0_addr = (uint32_t)pcm_left_data;
    dma_init_parameter.priority       = DMA_PRIORITY_HIGH;
    dma_single_data_mode_init(DMA1, DMA_CH2, &dma_init_parameter);
    /* connect DMA1_CH2 to HPDF_FLT1 */
    dma_channel_subperipheral_select(DMA1, DMA_CH2, DMA_SUBPERI7);
    /* enable DMA channel */
    dma_channel_enable(DMA1, DMA_CH2);
}
```

The data output by HPDF is played by I2S. Refer to the previous section, the audio sampling rate of I2S can be configured to 16KHz, the data channel is 16 bit, and the MSB standard mode is used. The I2S configuration is shown in ***Table 3-4. Configuration of I2S in GD project***.

Table 3-4. Configuration of I2S in GD project

```
/*!
    \brief        configure the I2S peripheral
    \param[in]    none
    \param[out] none
    \retval       none
*/
void i2s_config()
{
    spi_i2s_deinit(SPI1);
    /* I2S1 peripheral configuration */
    i2s_psc_config(SPI1,    I2S_AUDIOSAMPLE_16K,    I2S_FRAMEFORMAT_DT16B_CH16B,
    I2S_MCKOUT_ENABLE);
    i2s_init(SPI1, I2S_MODE_MASTERTX, I2S_STD_MSB, I2S_CKPL_HIGH);
    /* enable the I2S1 peripheral */
    i2s_enable(SPI1);
}
```

4. Revision history

Table 3-5. Revision history

Revision No.	Description	Date
1.0	Initial Release	Dec.13 2021

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.