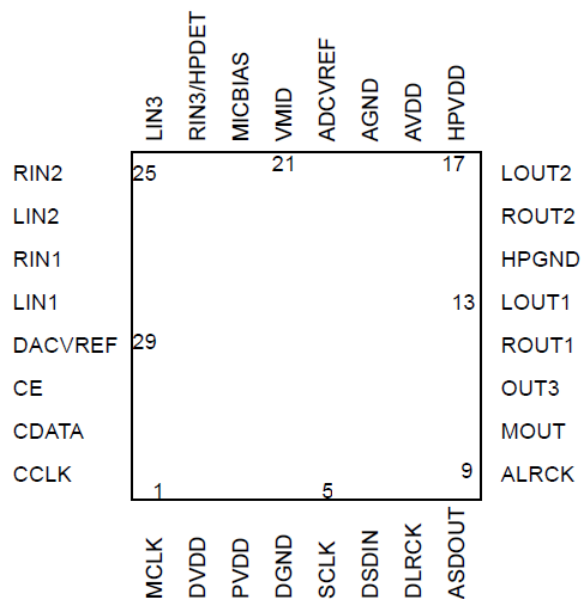


How to Program Driver Code of PA5750

DATE	Document	Version	Author
05-05-09	PA5750EVB Code	1.1	IGL

1. INTRODUCTION

PA5750 is a low power audio codec for portable audio application. The device supports standard SPI or I2C interface. External Micro-Controller can completely configure the device through writing to internal configuration registers.



The device can auto select SPI mode or I2C mode. To select SPI mode, apply high to low transition signal to CE pin. Otherwise, the device will operate in I2C mode.

In SPI mode, pin CE, CDATA, and CCLK functions as SPI_CS_n, SPI_MOSI, and SPI_CLK. Only “write” operation can be used in SPI mode.

In I2C mode, pin CE, CDATA, and CCLK functions as I2C_AD0, I2C_DAT, and I2C_CLK. “Write” and “read” operation all can be used in I2C mode.

- **SPI Timing diagram**

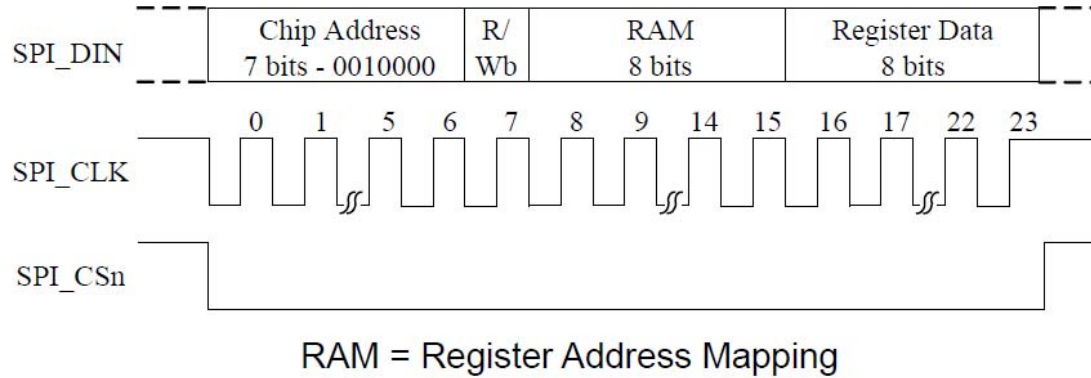


Figure 1 SPI Configuration Interface Timing Diagram

- **I2C Timing diagram**

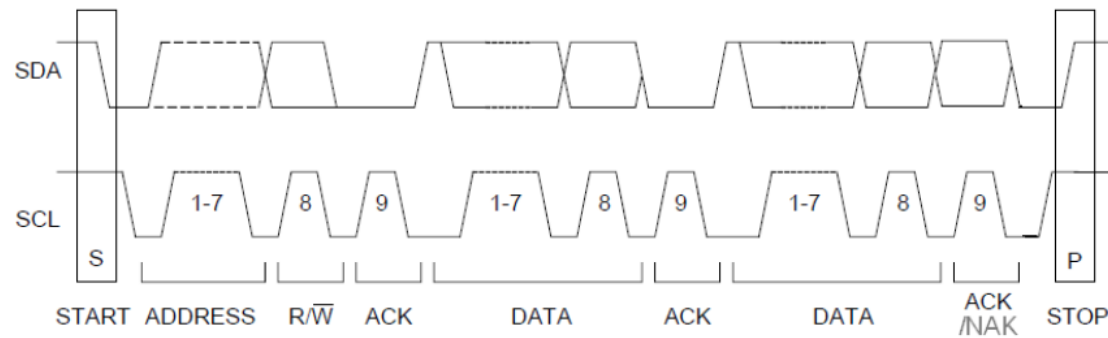


Figure 2 Complete Data Transfer for 2-wire Interface

- **Chip Address in SPI mode**

In SPI mode, the Chip address is “0010000” in binary, and R/W bit only can be 0 because only write operation can be used in SPI mode. So, the first byte in SPI timing only can be 0x20 in hexadecimal.

- **Chip Address in I2C mode**

Table 3 Write Data to Register in 2-wire Interface Mode

Chip Address	R/W	ACK	Register Address	Data to be written
001000	AD0	0	RAM	ACK DATA

Table 4 Read Data from Register in 2-wire Interface Mode

Chip Address	R/W	ACK	Register Address	
001000	AD0	0	RAM	

Chip Address	R/W	ACK	Data to be read	
001000	AD0	1	DATA	

In I2C mode, the Chip address is 0x20 for “write” operation and 0x21 for “read” operation if pin CE is tied to ground or low level. On the contrary, the chip address is 0x22 for “write” operation and 0x23 for “read” operation if pin CE is tied to PVDD.

2. Routine for Write and Read PA5750

- **Routine for Write PA5750**

```

/*****
** Function name:          WriteChip
**
** Descriptions:         Write 1 byte via SPI or I2C
**

```

```

** input parameters:
                UCHAR RAM, the Register Address
                UCHAR cData, the data written
** Returned value:    0, Write SUCCESSFUL
                    1, Write FAILED

```

```

**
** Used global variables: None
** Calling modules:      None

```

```

** Created by:          ProTek Analog
** Created Date:        2009/03/25

```

```

** -----
** Modified by:
** Modified date:
** -----

```

```

*****/

```

```

UCHAR WriteChip(UCHAR RAM,UCHAR cData);

```

- **Routine for Read PA5750**

```

/*****

```

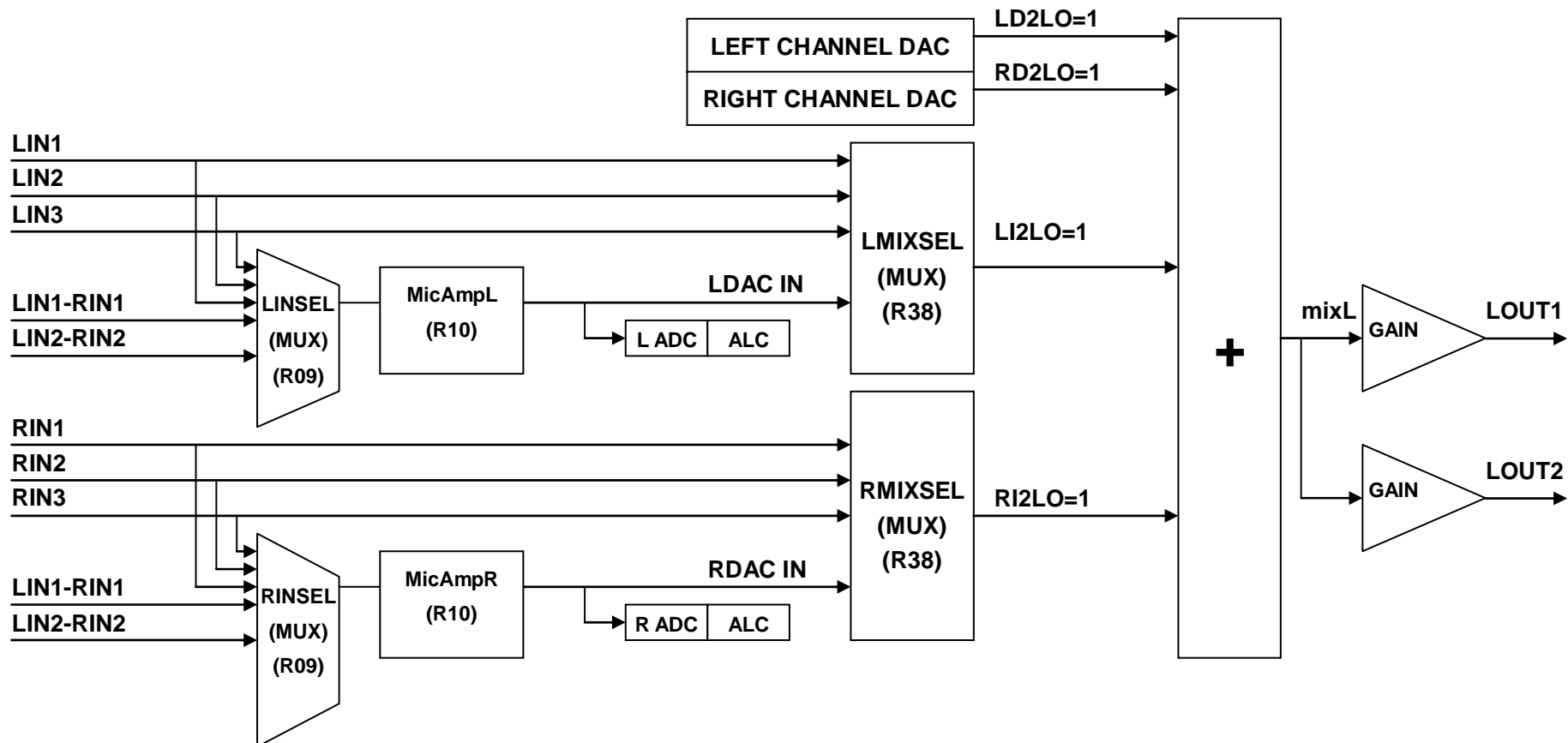
```

** Function name:      ReadChip
**
** Descriptions:      Read 1 byte via SPI or I2C
**
** input parameters:

```

```
                UCHAR RAM, the Register Address
** Returned value:      0, Read SUCESFUL,
                        1, Read FAILED
**
** Used global variables: None
** Calling modules:      None
**
** Created by:          ProTek Analog
** Created Date:        2009/03/25
** -----
** Modified by:
** Modified date:
** -----
*****/
UCHAR ReadChip(UCHAR RAM);
```

3. Driver for Bypass function of PA5750



```

void PA5750BYPass(void)
{
    //-----
    WriteChip(0x02,0xFF); //Disable DLL and Reset state machine

    WriteChip(0x08,0x00); //Set device to Slave mode.
                        //When device is in master mode, you can replace the "WriteChip(0x08,0x00)" with "WriteChip(0x08,0x80)"
    WriteChip(0x00,0x05); //set device in PLAYBACK & RECORD mode
    WriteChip(0x01,0x40); //pdn_ana=0,ibiasgen_pdn=0
    WriteChip(0x03,0xFC); //ADC Power-DOWN,DON'T USE MICROPHONE INPUT
    WriteChip(0x04,0xFF); //DAC Power-DOWN,ANALOG OUT ENABLE
    //-----
    //MIXER Register Map
    //-----
    WriteChip(0x26,0x09); //LMIXSEL=LIN2,RMIXSEL=RIN2

    WriteChip(0x27,0x50); //LMIXSEL TO Left MIXER,LMIXSEL VOLUME=0DB
    WriteChip(0x28,0x38);
    WriteChip(0x29,0x38);
    WriteChip(0x2A,0x50); //RMIXSEL TO RIGHT MIXER,RMIXSEL VOLUME=0DB
    WriteChip(0x2B,0x50); //LMIXSEL TO MONO MIXER,RMIXSEL VOLUME=0DB
    WriteChip(0x2C,0x38);

    WriteChip(0x2D,0x92); //ROUTINV=1,MONOINV=1,OUT3=VREF,VROI=1,DISABLE HEADPHONE SWITCH
    //-----

```

```
//SET VOLUME OF LOUT1/ROUT1/LOUT2/ROUT2/OUT3/MONOOOUT
```

```
//-----
```

```
WriteChip(0x2E,0X1E); //Step-up LO1 Volume
WriteChip(0x2F,0X1E); //Step-up RO1 Volume
WriteChip(0x30,0X1E); //Step-up LO2 Volume
WriteChip(0x31,0X1E); //Step-up RO2 Volume
WriteChip(0x32,0X1E); //Step-up MONO Volume
```

```
}
```

If Microphone and Micbias voltage of PA5750 are used in BYPASS mode, Register 0x03 must be set to 0x30. The code listed below is used for Microphone.

```
void PA5750BYPass(void)
```

```
{
```

```
//-----
```

```
WriteChip(0x02,0xFF); //Disable DLL and Reset state machine
```

```
WriteChip(0x08,0x00); //Set device to Slave mode.
```

```
//When device is in master mode, you can replace the "WriteChip(0x08,0x00)" with "WriteChip(0x08,0x80)"
```

```
WriteChip(0x00,0x05); //set device in PLAYBACK & RECORD mode
```

```
WriteChip(0x01,0x40); //pdn_ana=0,ibiasgen_pdn=0
```

```
WriteChip(0x03,0x30); //ADC Power-DOWN, USE MICROPHONE INPUT and Power up Micbias voltage
```

```
WriteChip(0x04,0xFF); //DAC Power-DOWN, ANALOG OUT ENABLE
```

```
WriteChip(0x09,0x88); //MIC BOOST GAIN =24DB
```

```
WriteChip(0x0A,0x00); //If Microphone is tied to LIN1&RIN1
```

```
//-----
```

```
//MIXER Register Map
```

```

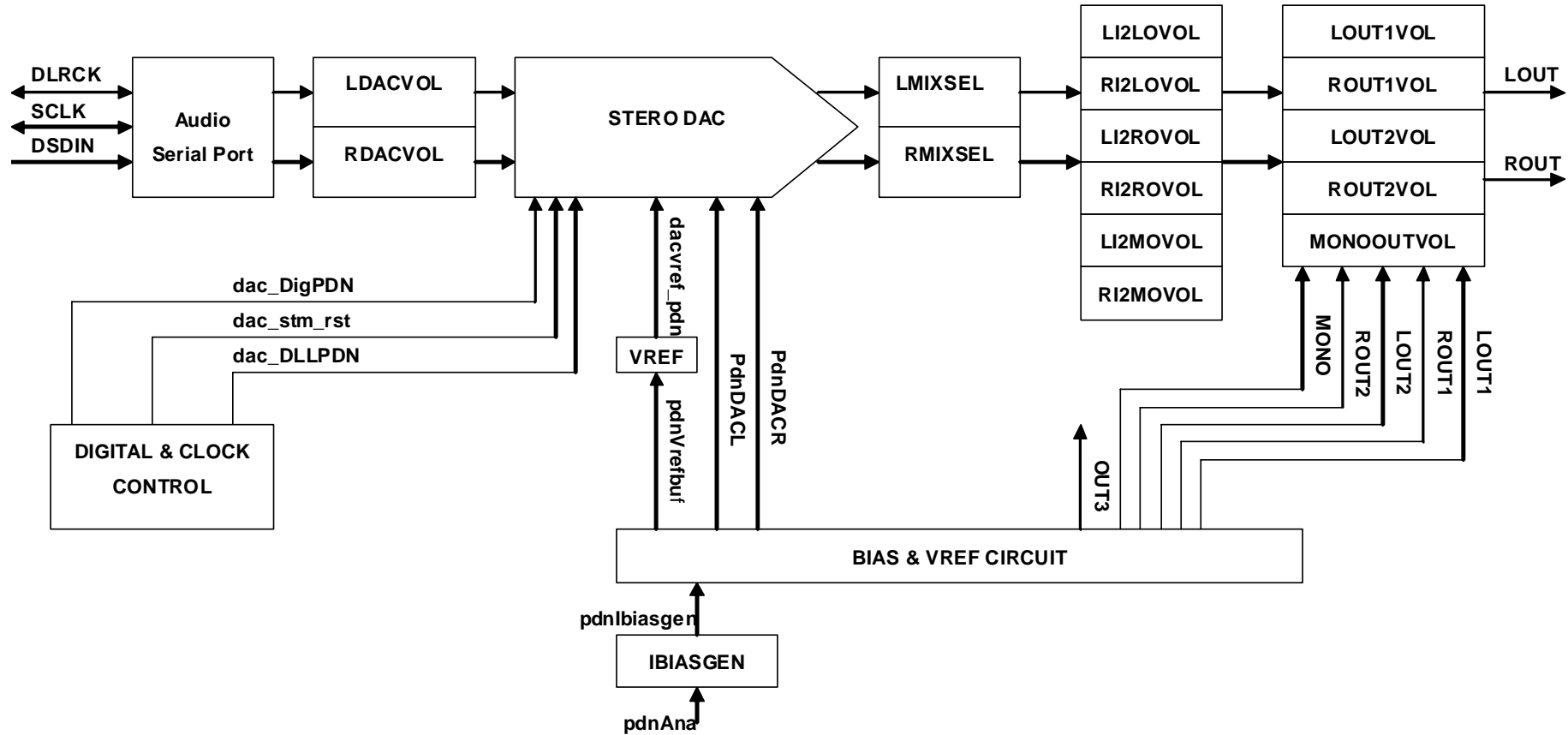
//-----
WriteChip(0x26,0x1B); //LMIXSEL=LDAC IN,RMIXSEL=RDAC IN

WriteChip(0x27,0x50); //LMIXSEL TO Left MIXER,LMIXSEL VOLUME=0DB
WriteChip(0x28,0x38);
WriteChip(0x29,0x38);
WriteChip(0x2A,0x50); //RMIXSEL TO RIGHT MIXER,RMIXSEL VOLUME=0DB
WriteChip(0x2B,0x50); //LMIXSEL TO MONO MIXER,RMIXSEL VOLUME=0DB
WriteChip(0x2C,0x38);

WriteChip(0x2D,0x92); //ROUTINV=1,MONOINV=1,OUT3=VREF,VROI=1,DISABLE HEADPHONE SWITCH
//-----
//SET VOLUME OF LOUT1/ROUT1/LOUT2/ROUT2/OUT3/MONOOOUT
//-----
WriteChip(0x2E,0X1E); //Step-up LO1 Volume
WriteChip(0x2F,0X1E); //Step-up RO1 Volume
WriteChip(0x30,0X1E); //Step-up LO2 Volume
WriteChip(0x31,0X1E); //Step-up RO2 Volume
WriteChip(0x32,0X1E); //Step-up MONO Volume
}

```

4. Driver for DAC function of PA5750



- **PA5750 operate in SLAVE mode**

In this example, the digital audio interface format is I2S-24bit, and the MCLK/LRCK ratio is 256.

The device operates in single speed mode.

```
void PA5750DACvoid)
```

```
{
    WriteChip(0x02,0xFF); // Disable DLL and Reset state machine
    WriteChip(0x08,0x00); //SET device to slave mode
    WriteChip(0x00,0x05); //set device to PLAY BACK AND RECORD MODE
    WriteChip(0x01,0x40); //pdn_ana=0,ibiasgen_pdn=0
    WriteChip(0x03,0xFF); //ADC POWER DOWN
    WriteChip(0x04,0x3F); //DAC & Analog Output Power-up
    //-----
    //DAC CONTROL Register Map
    //-----
    WriteChip(0x17,0x00); //I2S-24BIT
    WriteChip(0x18,0x02); //SINGLE SPEED,RATIO=256
    WriteChip(0x1A,0x00); //Left DAC Volume=0db
    WriteChip(0x1B,0x00); //Right DAC Volume=0db
    WriteChip(0x27,0xB8); //Left DAC TO Left MIXER
    WriteChip(0x28,0x38);
    WriteChip(0x29,0x38);
    WriteChip(0x2A,0xB8); //RIGHT DAC TO RIGHT MIXER
    WriteChip(0x2B,0xB8); //LEFT DAC TO MONO MIXER
    WriteChip(0x2C,0x38);
```

Driver Code for PA5750

```
WriteChip(0x2D,0x92); //ROUTINV=1,MONOINV=1,OUT3=VREF,VROI=1,DISABLE HEADPHONE SWITCH

WriteChip(0x2E,0X1E); //Step-up LO1 Volume
WriteChip(0x2F,0X1E); //Step-up RO1 Volume
WriteChip(0x30,0X1E); //Step-up LO2 Volume
WriteChip(0x31,0X1E); //Step-up RO2 Volume
WriteChip(0x32,0X1E); //Step-up MONO Volume

WriteChip(0x02,0xAA); //DAC ACTIVE,Very important
}
```

- **PA5750 operate in MASTER mode**

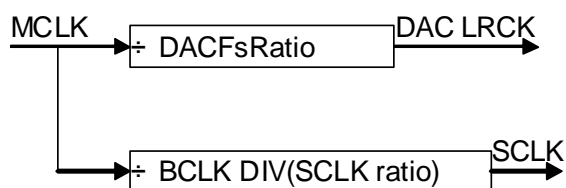
In MASTER mode, PA5750 can generate SCLK, DLRCK, ALRCK signal according to MCLK input and the setting of register 0x18. The table below explains the setting of Register 0x08 and Register 0x18.

Driver Code for PA5750

MCLK CLKDIV2=0	MCLK CLKDIV2=1	DAC Sample Rate (DLRCK)	DACFsRatio [4:0]	SCLK Ratio
Normal Mode				
12.288 MHz	24.576MHz	8 kHz (MCLK/1536)	01010	MCLK/6
		48 kHz (MCLK/256)	00010	MCLK/4
		12 kHz (MCLK/1024)	00111	MCLK/4
		16 kHz (MCLK/768)	00110	MCLK/6
		24 kHz (MCLK/512)	00100	MCLK/4
		32 kHz (MCLK/384)	00011	MCLK/6
		8 kHz (MCLK/1536)	01010	MCLK/4
		48 kHz (MCLK/256)	00010	MCLK/4
		96 kHz (MCLK/128)	00000	MCLK/2
11.2896 MHz	22.5792MHz	8.0182 kHz (MCLK/1408)	01001	MCLK/4
		44.1 kHz (MCLK/256)	00010	MCLK/4
		11.025 kHz (MCLK/1024)	00111	MCLK/4
		22.05 kHz (MCLK/512)	00100	MCLK/4
		8.0182 kHz (MCLK/1408)	01001	MCLK/4
		44.1 kHz (MCLK/256)	00010	MCLK/4
		88.2 kHz (MCLK/128)	00000	MCLK/2
18.432 MHz	36.864MHz	8 kHz (MCLK/2304)	01100	MCLK/6
		48 kHz (MCLK/384)	00011	MCLK/6
		12 kHz (MCLK/1536)	01010	MCLK/6
		16 kHz (MCLK/1152)	01000	MCLK/6
		24 kHz (MCLK/768)	00110	MCLK/6
		32 kHz (MCLK/576)	00101	MCLK/6
		8 kHz (MCLK/2304)	01100	MCLK/6
		48 kHz (MCLK/384)	00011	MCLK/6
		96 kHz (MCLK/192)	00001	MCLK/3
16.9344 MHz	33.8688MHz	8.0182 kHz (MCLK/2112)	01011	MCLK/6
		44.1 kHz (MCLK/384)	00011	MCLK/6
		11.025 kHz (MCLK/1536)	01010	MCLK/6
		22.05 kHz (MCLK/768)	00110	MCLK/6
		8.0182 kHz (MCLK/2112)	01011	MCLK/6
		44.1 kHz (MCLK/384)	00011	MCLK/6
		88.2 kHz (MCLK/192)	00001	MCLK/3
USB Mode				
12 MHz	24MHz	8 kHz (MCLK/1500)	11011	MCLK
		48 kHz (MCLK/250)	10010	MCLK
		8.0214 kHz (MCLK/1496)	11010	MCLK
		44.118 kHz (MCLK/272)	10011	MCLK
		11.0259 kHz (MCLK/1088)	11001	MCLK
		12 kHz (MCLK/1000)	11000	MCLK

Driver Code for PA5750

		16 kHz (MCLK/750)	10111	MCLK
		22.0588 kHz (MCLK/544)	10110	MCLK
		24 kHz (MCLK/500)	10101	MCLK
		32 kHz (MCLK/375)	10100*	MCLK
		8.0214 kHz (MCLK/1496)	11010	MCLK
		44.118 kHz (MCLK/272)	10011	MCLK
		8 kHz (MCLK/1500)	11011	MCLK
		48 kHz (MCLK/250)	10010	MCLK
		88.235 kHz (MCLK/136)	10001	MCLK
		96 kHz (MCLK/125)	10000	MCLK



6.1.9 Register 8 – Master Mode Control, Default 1000 0000

Bit Name	Bit	Description
MSC	7	0 – slave serial port mode 1 – master serial port mode (default)
MCLKDIV2	6	0 – MCLK not divide (default) 1 – MCLK divide by 2
BCLK_INV	5	0 – normal (default) 1 – BCLK inverted
BCLKDIV	4:0	00000 – master mode BCLK generated automatically based on the clock table (default) Others – MCLK/N, N=1~31

Driver Code for PA5750

BCLKDIV[4:0]	Frequency of SCLK
00000	According to Master Clock & Sample RateTable to generate SCLK
00001	MCLK/1
00010	MCLK/2
00011	MCLK/3
00100	MCLK/4
00101	MCLK/6
00110	MCLK/8
00111	MCLK/9
01000	MCLK/11
01001	MCLK/12
01010	MCLK/16
01011	MCLK/18
01100	MCLK/22
01101	MCLK/24
01110	MCLK/33
01111	MCLK/36
10000	MCLK/44
10001	MCLK/48
10010	MCLK/66
10011	MCLK/72
others	undefined

5. Driver for ADC function of PA5750

In this example, the digital audio interface format is I2S-24bit, and the MCLK/LRCK ratio is 256.

The device operates in single speed mode.

```
void PA5750ADC (void)
{
//-----
//Chip Control and Power Management Register Map
//-----
WriteChip(0x02,0xFF); // Disable DLL and Reset state machine
WriteChip(0x08,0x00); //Set device to Slave Mode
WriteChip(0x00,0x05); //set PA5750 to PLAYBACK & RECORD mode
WriteChip(0x01,0x40); //pdn_ana=0,ibiasgen_pdn=0
WriteChip(0x03,0x00); //ADC Power-up
//-----
//ADC Control Register Map
//-----
WriteChip(0x09,0x00); //ADC L/R PGA = 0dB
//-----
WriteChip(0x0A,0x50); //ADC input is LIN2/RIN2
//-----
WriteChip(0x0C,0x00); //I2S-24BIT
WriteChip(0x0D,0x02); //SINGLE SPEED,RATIO=256
```

Driver Code for PA5750

```
WriteChip(0x10,0x00); //ADC Left Volume=0db  
WriteChip(0x11,0x00); //ADC Right Volume=0db  
WriteChip(0x02,0x55); //START ADC DLL AND STATE MACHINE  
}
```