MUSIC TRIBE

i.MX RT105x I2S



JohnnyDu@musictribe.com

2018.10

Confidential. Not to be reproduced.

©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.

I2S 主要特性

- 带有帧同步功能的全双工音频接口，可支持比如I2S,AC97,TDM及Codec/DSP等接口
- 发送和接收具有独立的位时钟和帧同步，SAI1支持4条数据线，SAI2~3各一条数据线
- 每个数据线可支持的的最大帧长度为32个words
- Word大小可配置为8,16,24或32bits
- 可分别配置第一个Word和其余Word的大小
- 每个发送和接收的数据通道各包含一个32 x 32bit 的异步FIFO
- 支持FIFO出错后自动及稳定重启，无需软件干预
- 支持将8bit和16bit数据打包到每个32bit的FIFO中
- 支持将多个数据选的FIFO组合到单个数据线的FIFO
- 包含运行模式，停止模式，低泄露模式和调试模式四种操作模式

Learn. Teach. Inspire.

Confidential. Not to be reproduced.
Confidential. Not to be reproduced.
©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.
©2016 MUSIC. All rights reserved.

I2S 框图

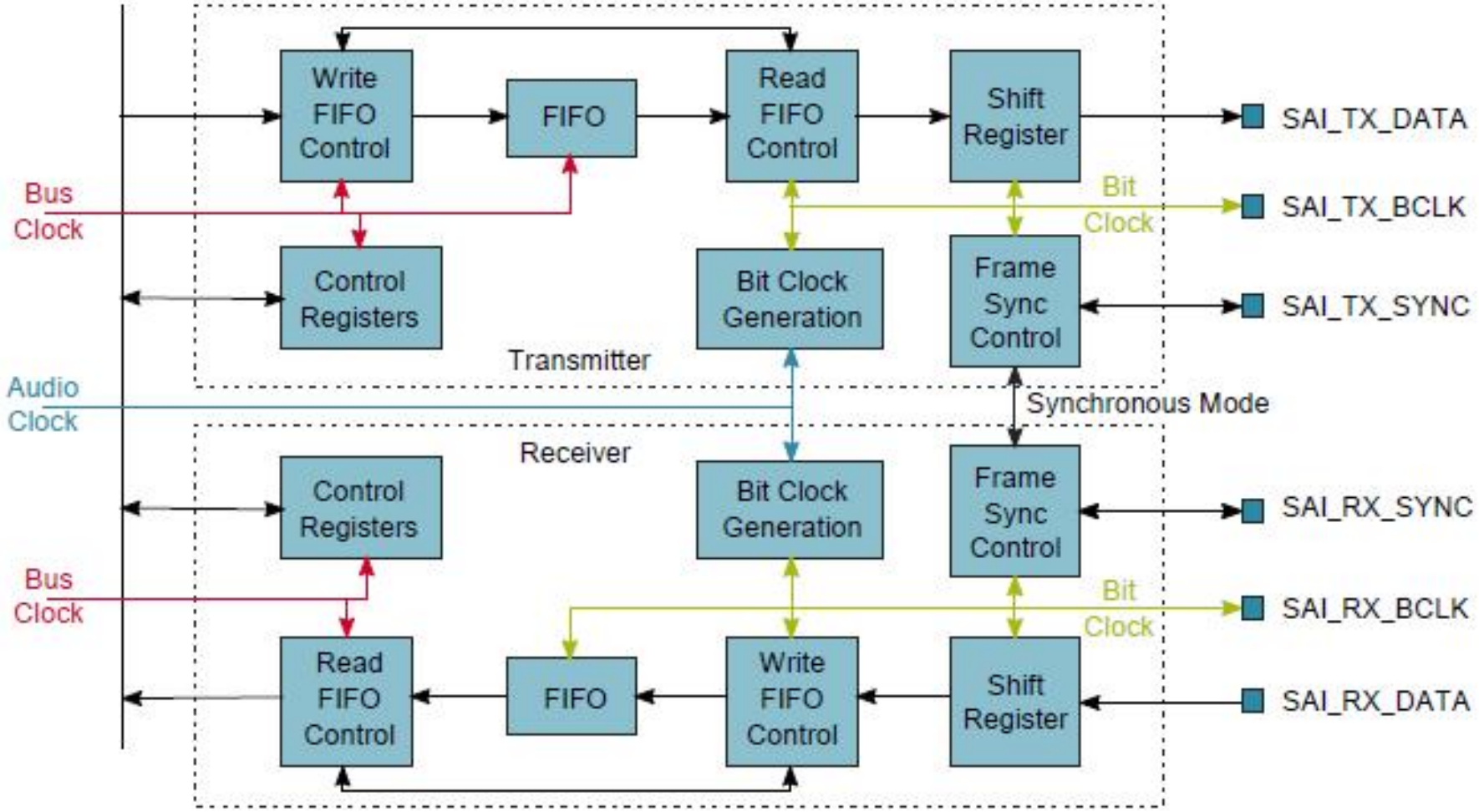


Figure 48-1. I²S/SAI block diagram

Learn. Teach. Inspire.

Confidential. Not to be reproduced.
©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.
Confidential. Not to be reproduced.
©2016 MUSIC All rights reserved.

I2S 外部信号

名称	功能	I/O
SAI_TX_BCLK	发送位时钟，由外部产生时作输入，由内部产生时作输出	I/O
SAI_TX_SYNC	发送帧同步，用于发送位时钟的采样同步，由外部产生时为输入，由内部产生时为输出	I/O
SAI_TX_DATA[3:0]	发送数据，通过位时钟同步采样数据，不传输数据时端口为三态	O
SAI_RX_BCLK	接收位时钟，由外部产生时作输入，由内部产生时作输出	I/O
SAI_RX_SYNC	接收帧同步，用于接收位时钟的采样同步，由外部产生时为输入，由内部产生时为输出	I/O
SAI_RX_DATA[3:0]	接收数据，通过位时钟同步采样数据	I
MCLK_EN	主时钟	I

Learn. Teach. Inspire.

Confidential. Not to be reproduced.
Confidential. Not to be reproduced.
©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.
©2016 MUSIC. All rights reserved.

主时钟：

当发送或接收的位时钟由内部产生时，主时钟用于产生位时钟。发送和接收可各自选择总线时钟和最多3个主时钟用于产生位时钟。

位时钟和帧同步：

SAI的发送和接收支持异步各自的位时钟和帧同步，可由内部产生也可外部提供。可选择在接收和发送或多个SAI外设之间同步位时钟和帧时钟。

- 发送和接收都为异步模式，发送和接收使用各自的位时钟和帧同步
- 发送为异步接收为同步，发送和接收将使用发送的位时钟和帧同步
- 发送为同步接收为异步，发送和接收将使用接收的位时钟和帧同步

Learn. Teach. Inspire.

Confidential. Not to be reproduced.
Confidential. Not to be reproduced.
©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.
©2016 MUSIC. All rights reserved.

数据FIFO：

每个发送和接收通道各包含一个32x32bit的FIFO，使用SAI发送/接收数据寄存器自动访问FIFO数据。

数据对齐：

可通过配置First Bit Shifted(起始位索引，可选0~31)使FIFO中的数据在任意位置对齐

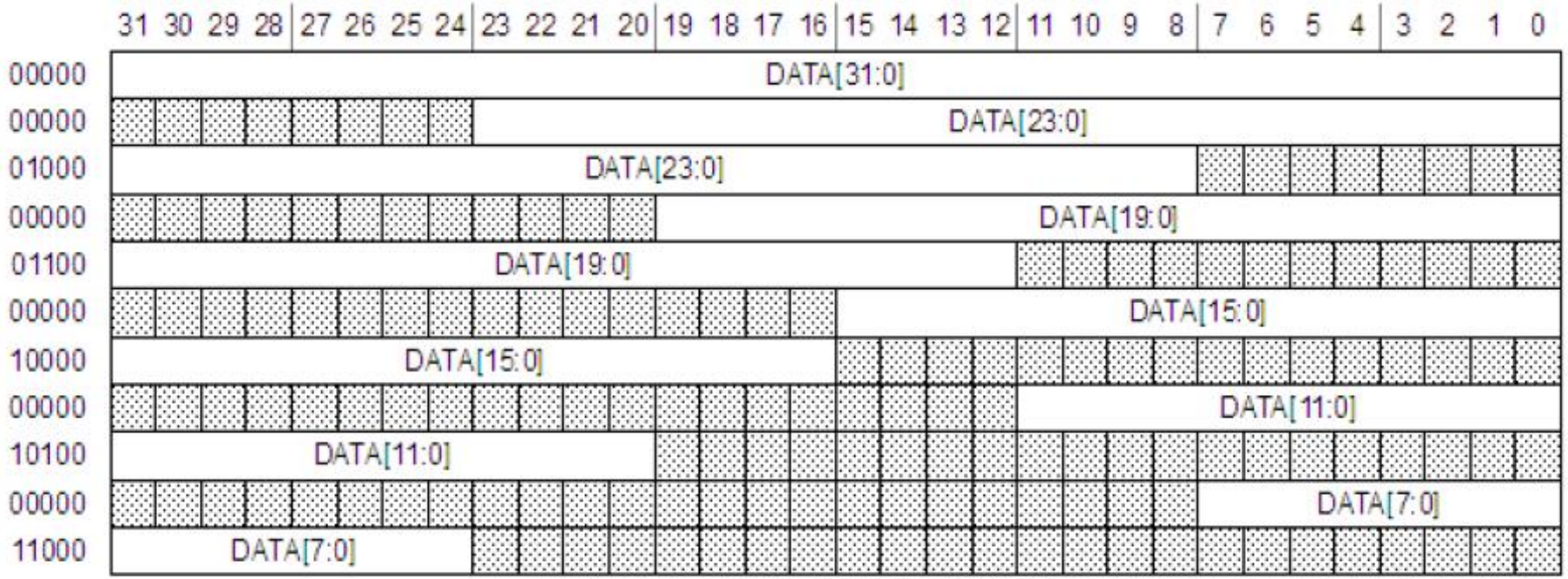


Figure 48-2. SAI first bit shifted, LSB first

Confidential. Not to be reproduced.
Confidential. Not to be reproduced.
©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.
©2016 MUSIC All rights reserved.

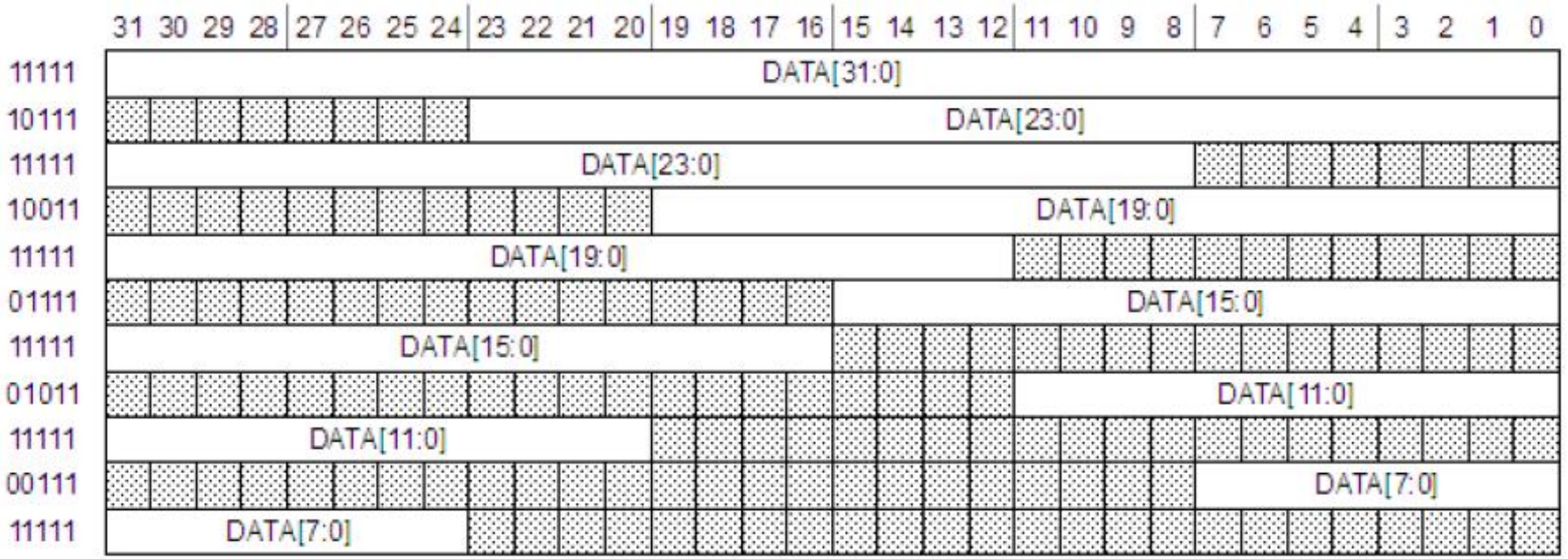


Figure 48-3. SAI first bit shifted, MSB first

例：

1，MSB First，FBT为11111，Word宽度为32bit，最高位处于bit 31处

2，MSB First，FBT为10111，Word宽度为24bit，最高位处于bit 23处

3，MSB First，FBT为01111，Word宽度为16bit，最高位处于bit 15处

Learn. Teach. Inspire.

Confidential. Not to be reproduced.
©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.
Confidential. Not to be reproduced.
©2016 MUSIC. All rights reserved.

```
#define NUM_SLOTS                2      // 一个FS分为两个slot
#define NUM_SAMPLES              4      // 4个FS为一个block

typedef struct
{
    // 4个样本点接收及发送Buff，由两个Ping-Pong buffer构成
    DATA_UNION  TxIntBuff[2][NUM_SLOTS*NUM_SAMPLES];
    DATA_UNION  RxIntBuff[2][NUM_SLOTS*NUM_SAMPLES];

    // 指向 RxIntBuff及TxIntBuff 的索引
    uint8_t     RxIntBuffIdx;
    uint8_t     TxIntBuffIdx;
}I2S_DATA;

// 要使用Freescale提供的库函数，需分别定义一个发送和接收的处理句柄和驱动层的回掉函数
sai_handle_t I2sRxHandle = {0};
sai_handle_t I2sTxHandle = {0};
void I2sTxCallback(I2S_Type *base, sai_handle_t *handle, status_t status, void *userData);
void I2sRxCallback(I2S_Type *base, sai_handle_t *handle, status_t status, void *userData);
```

Learn. Teach. Inspire.

Confidential. Not to be reproduced.

©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.

Confidential. Not to be reproduced.

©2016 MUSIC All rights reserved.

```
#define SAI_MCLK_CONFIG          (IOMUXC_SW_PAD_CTL_PAD_SRE(0)    | \
                                  IOMUXC_SW_PAD_CTL_PAD_DSE(6)    | \
                                  IOMUXC_SW_PAD_CTL_PAD_SPEED(2)  | \
                                  IOMUXC_SW_PAD_CTL_PAD_ODE(0)    | \
                                  IOMUXC_SW_PAD_CTL_PAD_PKE(1)    | \
                                  IOMUXC_SW_PAD_CTL_PAD_PUE(0)    | \
                                  IOMUXC_SW_PAD_CTL_PAD_PUS(0)    | \
                                  IOMUXC_SW_PAD_CTL_PAD_HYS(0))
```

其它端口配置相同，此处不一一列出

```
// I2S 端口配置
void I2sIoConfig(void)
{
    IOMUXC_SetPinMux(IOMUXC_GPIO_AD_B1_09_SAI1_MCLK, 0U);

    IOMUXC_SetPinMux(IOMUXC_GPIO_AD_B1_10_SAI1_RX_SYNC, 0U);
    IOMUXC_SetPinMux(IOMUXC_GPIO_AD_B1_11_SAI1_RX_BCLK, 0U);
    IOMUXC_SetPinMux(IOMUXC_GPIO_AD_B1_12_SAI1_RX_DATA00, 0U);
```

Learn. Teach. Inspire.

Confidential. Not to be reproduced.

©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.

Confidential. Not to be reproduced.

©2016 MUSIC All rights reserved.

```
    IOMUXC_SetPinMux(IOMUXC_GPIO_AD_B1_13_SAI1_TX_DATA00, 0U);
    IOMUXC_SetPinMux(IOMUXC_GPIO_AD_B1_14_SAI1_TX_BCLK, 0U);
    IOMUXC_SetPinMux(IOMUXC_GPIO_AD_B1_15_SAI1_TX_SYNC, 0U);

    IOMUXC_SetPinConfig(IOMUXC_GPIO_AD_B1_09_SAI1_MCLK, SAI_MCLK_CONFIG);

    IOMUXC_SetPinConfig(IOMUXC_GPIO_AD_B1_10_SAI1_RX_SYNC, SAI_RX_FS_CONFIG);
    IOMUXC_SetPinConfig(IOMUXC_GPIO_AD_B1_11_SAI1_RX_BCLK, SAI_RX_BCLK_CONFIG);
    IOMUXC_SetPinConfig(IOMUXC_GPIO_AD_B1_12_SAI1_RX_DATA00,
                        SAI_RX_DATA0_CONFIG);

    IOMUXC_SetPinConfig(IOMUXC_GPIO_AD_B1_15_SAI1_TX_SYNC, SAI_TX_FS_CONFIG);
    IOMUXC_SetPinConfig(IOMUXC_GPIO_AD_B1_14_SAI1_TX_BCLK, SAI_TX_BCLK_CONFIG);
    IOMUXC_SetPinConfig(IOMUXC_GPIO_AD_B1_13_SAI1_TX_DATA00,
                        SAI_TX_DATA0_CONFIG);

    // Enable MCLK output
    IOMUXC_GPR->GPR1 |= IOMUXC_GPR_GPR1_SAI1_MCLK_DIR_MASK;
}
```

Learn. Teach. Inspire.

Confidential. Not to be reproduced.
©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.
Confidential. Not to be reproduced.
©2016 MUSIC. All rights reserved.

```
// I2S 配置

void I2sConfig(void)
{
    uint32_t mclkSourceClockHz = 0U;
    sai_config_t config;
    sai_transfer_t xfer;
    sai_transfer_format_t format;

    // 接收为异步
    SAI_RxGetDefaultConfig(&config);
    config.syncMode = kSAI_ModeAsync;
    SAI_RxInit(SAI1, &config);

    // 发送为同步，使用接收的BCLK和FS
    SAI_TxGetDefaultConfig(&config);
    config.syncMode = kSAI_ModeSync;
    SAI_TxInit(SAI1, &config);
```

Learn. Teach. Inspire.

Confidential. Not to be reproduced.

©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.

Confidential. Not to be reproduced.

©2016 MUSIC All rights reserved.

```
// 48K FS, Word width 32bits, 12.288M MCLK, I2S mode, 64 bit clock
// 两个watermark(2*4=8, 8个字节, 即一个FS)产生一次中断
format.sampleRate_Hz = kSAI_SampleRate48KHz;
format.bitWidth = kSAI_WordWidth32bits;
format.stereo = kSAI_Stereo;
format.masterClockHz = 12288000U;
format.watermark = 2;
format.channel = 0U;
format.protocol = config.protocol;
format.isFrameSyncCompact = false;

// 设置发送和接收的处理句柄
SAI_TransferTxCreateHandle(SAI1, &I2sTxHandle, I2sTxCallback, NULL);
SAI_TransferRxCreateHandle(SAI1, &I2sRxHandle, I2sRxCallback, NULL);

// 设置发送和接收的I2S格式
mclkSourceClockHz = format.masterClockHz;
SAI_TransferTxSetFormat(SAI1, &I2sTxHandle, &format, mclkSourceClockHz,
                        format.masterClockHz);
SAI_TransferRxSetFormat(SAI1, &I2sRxHandle, &format, mclkSourceClockHz,
                        format.masterClockHz);
```

Learn. Teach. Inspire.

Confidential. Not to be reproduced.

©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.

Confidential. Not to be reproduced.

©2016 MUSIC. All rights reserved.

```
    // 32个字节, 每接收32个字节(4个FS)调用一次callback函数
    xfer.dataSize = 4 * NUM_SLOTS * NUM_SAMPLES;
    xfer.data = I2sData.TxIntBuff[I2sData.TxIntBuffIdx][0].ByteData;
    SAI_TransferSendNonBlocking(SAI1, &I2sTxHandle, &xfer);

    // 32个字节, 每接收32个字节(4个FS)调用一次callback函数
    xfer.data = I2sData.RxIntBuff[I2sData.RxIntBuffIdx][0].ByteData;
    SAI_TransferReceiveNonBlocking(SAI1, &I2sRxHandle, &xfer);
}


// I2S 发送完成 callback函数
void I2sTxCallback(I2S_Type *base, sai_handle_t *handle, status_t status, void *userData)
{
    sai_transfer_t xfer;

    if(kStatus_SAI_RxError == status)
    {
        /* Handle the error. */
    }
```

Learn. Teach. Inspire.

Confidential. Not to be reproduced.

©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.

Confidential. Not to be reproduced.

©2016 MUSIC All rights reserved.

```
    else
    {
        I2sData.TxIntBuffIdx = 1 - I2sData.TxIntBuffIdx;

        // 发送32个字节, 4个FS, 4个立体声样本点
        xfer.dataSize = 4 * NUM_SLOTS * NUM_SAMPLES;
        xfer.data = I2sData.TxIntBuff[I2sData.TxIntBuffIdx][0].ByteData;
        SAI_TransferSendNonBlocking(SAI1, &I2sTxHandle, &xfer);
    }
}


// I2S 接收完成 callback函数
void I2sRxCallback(I2S_Type *base, sai_handle_t *handle, status_t status, void *userData)
{
    uint8_t i;
    sai_transfer_t xfer;

    if(kStatus_SAI_RxError == status)
    {
        /* Handle the error. */
    }
```

Learn. Teach. Inspire.

Confidential. Not to be reproduced.

©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.

Confidential. Not to be reproduced.

©2016 MUSIC All rights reserved.

```
    else
    {
        memcpy(I2sData.TxIntBuff[I2sData.RxIntBuffIdx],
               I2sData.RxIntBuff[I2sData.RxIntBuffIdx],
               (NUM_SLOTS * NUM_SAMPLES * sizeof(int32_t)));

        I2sData.RxIntBuffIdx = 1 - I2sData.RxIntBuffIdx;

        // 接收32个字节, 4个FS, 4个立体声样本点
        xfer.dataSize = 4 * NUM_SLOTS * NUM_SAMPLES;
        xfer.data = I2sData.RxIntBuff[I2sData.RxIntBuffIdx][0].ByteData;
        SAI_TransferReceiveNonBlocking(SAI1, &I2sRxHandle, &xfer);
    }
}
```


Confidential. Not to be reproduced.
Confidential. Not to be reproduced.
©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.
©2016 MUSIC All rights reserved.



Confidential. Not to be reproduced.
©2018 MUSIC Tribe Global Brands Ltd. All rights reserved.