



GLOBAL **EDGE**
Intelligence Of Things™

I2S – Inter-IC Sound

Akshata Mukund Shetty

Contents

- Introduction
- Modes of I2S module
- Features
- Example
- I2S configuration API's

Introduction

- I²S (Inter-IC Sound) is an electrical serial bus interface standard used for connecting digital audio devices together.
- The bus can handle audio data.
- A 3-line serial bus is used consisting of a line for two time-multiplexed data channels, a word select line and a clock line.

Introduction

The bus has three lines:

- continuous serial clock (SCK);
- word select (WS);

WS = 0; channel 1 (left);

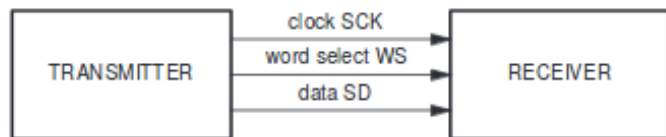
WS = 1; channel 2 (right).

- serial data (SD);

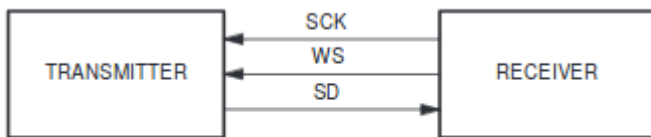
serial data (SD) can be called SDATA, SDIN, SDOUT, DACDAT, ADCDAT, etc

The device generating SCK and WS is the master.

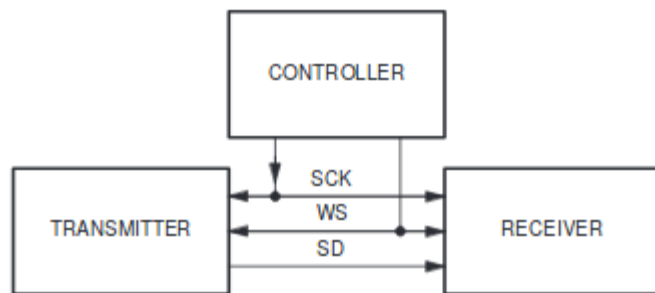
Introduction



TRANSMITTER = MASTER



RECEIVER = MASTER



CONTROLLER = MASTER

Different modes of I2S module

The I2S module has three modes:

- Master
- Controller
- Slave

Features

- 8 - 32 data bits per sample
- 16, 32, 48, or 64 bit word select period
- Data rate up to 192 kHz with 64 bit word select period: 12.288 MHz.
- Tx and Rx FIFO interrupts
- DMA support
- Independent left and right channel FIFOs
- Independent enable of Rx and Tx

Example of I2S configuration

```
#include "driver/i2s.h"
#include "freertos/queue.h"

static const int i2s_num = 0; // i2s port number

static const i2s_config_t i2s_config = {
    .mode = I2S_MODE_MASTER | I2S_MODE_TX,
    .sample_rate = 44100,
    .bits_per_sample = 16,
    .channel_format = I2S_CHANNEL_FMT_RIGHT_LEFT,
    .communication_format = I2S_COMM_FORMAT_I2S | I2S_COMM_FORMAT_I2S_MSB,
    .intr_alloc_flags = 0, // default interrupt priority
    .dma_buf_count = 8,
    .dma_buf_len = 64,
    .use_apll = 0
};
```


Example of I2S configuration

```
static const i2s_pin_config_t pin_config = {  
    .bck_io_num = 26,  
    .ws_io_num = 25,  
    .data_out_num = 22,  
    .data_in_num = I2S_PIN_NO_CHANGE  
};  
  
...  
i2s_driver_install(i2s_num, &i2s_config, 0, NULL); //install and  
start i2s driver  
  
i2s_set_pin(i2s_num, &pin_config);  
  
i2s_set_sample_rates(i2s_num, 22050); //set sample rates  
  
i2s_driver_uninstall(i2s_num); //stop & destroy i2s driver
```

I2S configuration API's

- `void I2S_init(void)` : Enables the I2S interface
- `void I2S_start(void)` : Starts the I2S interface.
- `void I2S_stop(void)` : Disables the I2S interface.
- `void I2S_enableTx(void)` : Enables the Tx direction of the I2S interface.
- `void I2S_disableTx(void)` : Disables the Tx direction of the I2S interface.
- `void I2S_enableRx(void)` : Enables the Rx direction of the I2S interface.
- `void I2S_disableRx(void)` : Disables the Rx direction of the I2S interface.

I2S configuration API's

- `uint8 I2S_readRxStatus(void)` : Returns state in the I2S Rx status register
- `uint8 I2S_readTxStatus(void)` : Returns state in the I2S Tx status register
- `uint8 I2S_readByte(uint8 word_Select)` : Returns a single byte from the Rx FIFO.
- `void I2S_writeByte(uint8 wrData, uint8 word_Select)` : Writes a single byte into the Tx FIFO.
- `void I2S_clearRxFIFO(void)` : Clears out the Rx FIFO.
- `void I2S_clearTxFIFO(void)` : Clears out the Tx FIFO.
- `void I2S_sleep(void)` : Saves configuration and disables the I2S interface
- `void I2S_wakeUp(void)` : Restores configuration and enables the I2S interface

*Large enough to Deliver, **Small enough to Care***



Global Village
IT SEZ
Bangalore



South Main Street
Milpitas
California

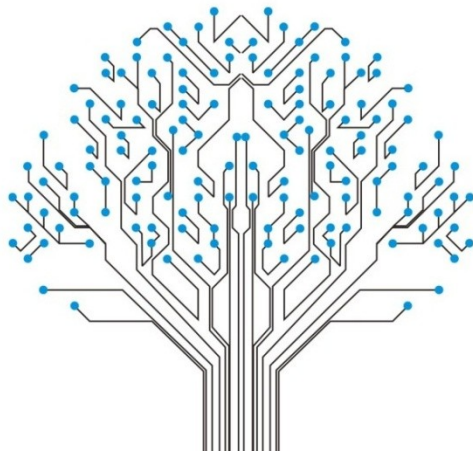
Raheja Mindspace
IT Park
Hyderabad



www.globaledgeoft.com



Thank you



Fairness

Learning

Responsibility

Innovation

Respect