

# Infineon Arduino Library Documentation

**Author:** Infineon Technologies AG

**Date:** August 14, 2018

# Contents

<b>1 Hierarchical Index</b>	<b>3</b>
1.1 Class Hierarchy . . . . .	3
<b>2 Class Index</b>	<b>4</b>
2.1 Class List . . . . .	4
<b>3 Class Documentation</b>	<b>5</b>
3.1 XMCI2SClass Class Reference . . . . .	5
3.1.1 Constructor & Destructor Documentation . . . . .	6
3.1.1.1 XMCI2SClass() . . . . .	6
3.1.2 Member Function Documentation . . . . .	6
3.1.2.1 begin() . . . . .	7
3.1.2.2 configureInterrupt() . . . . .	7
3.1.2.3 read() . . . . .	7
<b>Index</b>	<b>8</b>

---

# 1 Hierarchical Index

## 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

SignalInput	
XMCI2SClass . . . . .	5

---

## 2 Class Index

### 2.1 Class List

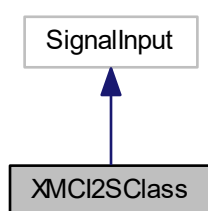
Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">XMCI2SClass</a> . . . . .	5
---------------------------------------	---

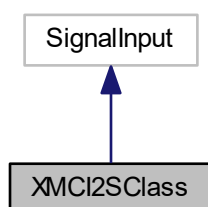
## 3 Class Documentation

### 3.1 XMCI2SClass Class Reference

Inheritance diagram for XMCI2SClass:



Collaboration diagram for XMCI2SClass:



#### Public Member Functions

- **XMCI2SClass** (bool isMono=true, uint8\_t bitsPerSample=16)  
*Constructor of the I2S class.*
- void **begin** ()  
*Starts I2S communication with default configuration.*
- void **begin** (uint32\_t samplingRate, uint8\_t frameLength=32, uint8\_t dataDelay=1, uint8\_t dataBits=16)  
*Starts I2S communication.*
- void **end** ()  
*Ends I2S communication with default configuration.*
- int **peek** ()

*Returns the latest received sample.*

- void `read` (int16\_t \*readBuffer, uint16\_t readSize)

*Copies samples into an int array.*

- void `enableMasterClock` ()

*Enables master clock output signal with a fixed phase relation to the shift clock to support oversampling for audio components.*

- uint16\_t `available` ()

*Returns the number of samples in buffer available for read.*

- void `onSampleReceived` ()

*Read in one sample on rising/falling edge of WA signal.*

- void `downScaleByTwo` ()

*Reduces the sampling frequency by half; this function can be used to sample at below the lowest frequency of the microphone.*

- void `configureInterrupt` (interrupt\_cb\_t callback)

*Configures the interrupt generated when the buffer for audio samples is full.*

## Static Public Member Functions

- static void `dmaCallback` ()

*Wrapper function for custom callback. Toggles the `_inDmaInProgress` flag.*

## 3.1.1 Constructor & Destructor Documentation

### 3.1.1.1 XMCI2SClass()

```
XMCI2SClass::XMCI2SClass (
    bool isMono = true,
    uint8_t bitsPerSample = 16 )
```

Constructor of the I2S class.

#### Parameters

<i>isMono</i>	if mono channel is enabled and is set to true by default. If you are using the Adafruit microphone breakout, this corresponds to an unconnected SEL.
---------------	--

## 3.1.2 Member Function Documentation

### 3.1.2.1 begin()

```
void XMCI2SClass::begin (
    uint32_t samplingRate,
    uint8_t frameLength = 32,
    uint8_t dataDelay = 1,
    uint8_t dataBits = 16 )
```

Starts I2S communication.

#### Parameters

<i>samplingRate</i>	samples per seconds
<i>dataBits</i>	length of one data frame. Ranges from 1 to 16 (size of the FIFO buffer)
<i>frameLength</i>	number of bits between 2 changing edges
<i>dataDelay</i>	the number of clock cycles after which a transmission of one samples is started

### 3.1.2.2 configureInterrupt()

```
void XMCI2SClass::configureInterrupt (
    interrupt_cb_t callback )
```

Configures the interrupt generated when the buffer for audio samples is full.

#### Parameters

<i>callback</i>	the callback where data processing happens.
-----------------	---

### 3.1.2.3 read()

```
void XMCI2SClass::read (
    int16_t * readBuffer,
    uint16_t readSize )
```

Copies samples into an int array.

#### Parameters

<i>readBuffer</i>	the user-defined buffer of integers.
<i>readSize</i>	number of samples that should be copied into the buffer. Must be smaller than the buffer's length

## Index

begin

XMCI2SClass, [6](#)

configureInterrupt

XMCI2SClass, [7](#)

read

XMCI2SClass, [7](#)

XMCI2SClass, [5](#)

begin, [6](#)

configureInterrupt, [7](#)

read, [7](#)

XMCI2SClass, [6](#)



#### Trademarks of Infineon Technologies AG

μHVIC™, μIPM™, μPFC™, AU-ConvertIR™, AURIX™, C166™, CanPAK™, CIPOS™, CIPURSE™, CoolDP™, CoolGaN™, COOLiR™, CoolMOS™, CoolSET™, CoolSiC™, DAVE™, DI-POL™, DirectFET™, DrBlade™, EasyPIM™, EconoBRIDGE™, EconoDUAL™, EconoPACK™, EconoPIM™, EiceDRIVER™, eupec™, FCOS™, GaNpowIR™, HEXFET™, HITFET™, HybridPACK™, iMOTION™, IRAM™, ISOFACE™, IsoPACK™, LEDriviR™, LITIX™, MIPAQ™, ModSTACK™, my-d™, NovalithiC™, OPTIGA™, OptiMOS™, ORIGA™, PowIRaudio™, PowIRstage™, PrimePACK™, PrimeSTACK™, PROFET™, PRO-SiL™, RASIC™, REAL3™, SmartLEWIS™, SOLID FLASH™, SPOC™, StrongIRFET™, SuplIRBuck™, TEMPFET™, TRENCHSTOP™, TriCore™, UHVIC™, XHP™, XMC™.

Trademarks Update 2015-12-22

#### Other Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

#### IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

#### WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury